

A THREE-DIMENSIONAL GROUND-WATER-FLOW MODEL MODIFIED  
TO REDUCE COMPUTER-MEMORY REQUIREMENTS AND BETTER SIMULATE  
CONFINING-BED AND AQUIFER PINCHOUTS

By P. Patrick Leahy

---

U. S. GEOLOGICAL SURVEY

Water-Resources Investigations 82-4023



Trenton, New Jersey

1982

UNITED STATES DEPARTMENT OF THE INTERIOR

JAMES G. WATT, Secretary

GEOLOGICAL SURVEY

Dallas L. Peck, Director

---

For additional information  
write to:

U.S. Geological Survey  
Room 430, Federal Building  
402 East State Street  
Trenton, New Jersey 08608

Copies of this report can be  
purchased from:

Open-File Services Section  
Western Distribution Branch  
U.S. Geological Survey  
Box 25425, Federal Center  
Lakewood, Colorado 80225  
(Telephone: [303] 234-5888)

## CONTENTS

	Page
Abstract.....	1
Introduction.....	1
Memory reduction.....	2
Simulating confining bed and aquifer pinchouts.....	6
Input requirements.....	8
Comparison of memory requirements and cost for simulation of the regional aquifer system.....	9
Conclusions.....	12
References.....	12
Appendix 1: Data deck instructions.....	14
Appendix 2: Listing of source code.....	23

## ILLUSTRATIONS

Figure 1.--Map showing location of northern Atlantic Coastal Plain.....	3
2.--Numbering scheme for finite-difference mesh used by the Trescott (1975) model and the modified model.....	5
3.--Generalized cross sections and conceptualizations of aquifer and confining-bed pinchouts.....	7
4.--Finite-difference grid used to discretize the northern Atlantic Coastal Plain aquifer system.	10

## TABLES

Table 1.--Cost savings realized by modified model for sample problems.....	12
---	----

A THREE-DIMENSIONAL GROUND-WATER-FLOW MODEL MODIFIED TO REDUCE  
COMPUTER-MEMORY REQUIREMENTS AND BETTER SIMULATE  
CONFINING-BED AND AQUIFER PINCHOUTS

By P. Patrick Leahy

ABSTRACT

The Trescott (1975) computer program for modeling ground-water flow in three dimensions has been modified to (1) treat confining-bed and aquifer pinchouts more realistically and (2) reduce the computer-memory requirements needed for the input data. Simulating aquifer systems having nonrectangular external boundaries with the original program may result in many nodes that are not involved in the numerical solution of the problem, but that require computer memory.

The Trescott program and the modified program are used to develop a cost comparison of the computer programs for large field problems. Steady-state simulations of the northern Atlantic Coastal Plain regional aquifer system and the New Jersey Coastal Plain are used. The comparison shows a 30 percent cost savings for the regional model, which has 11,560 unused nodes of a total of 27,200. The New Jersey model shows an 8 percent savings, where 1,540 of the 14,790 nodes are unused.

Appendices provide data-deck instructions and a listing of the FORTRAN source code.

INTRODUCTION

A computer model (Trescott, 1975) for simulating ground-water flow in three dimensions was developed by the U.S. Geological Survey. Hydrologists of the Survey have applied this program to numerous field problems. The structure of the FORTRAN code, however, is inconsistent with efficient and cost-effective simulation of certain physical and hydrologic situations. In particular, confining-bed and aquifer pinchouts are not easily simulated with the Trescott model. Also, simulating complex external boundaries (shape of the entire aquifer system) results in inefficient use of computer storage. Developing a model that effectively handles these situations is an outgrowth of the study of the northern Atlantic Coastal Plain aquifer system begun in 1979 by the Geological Survey. This project is part of the Regional Aquifer System Analysis (RASA) program that began in 1977. These studies define the geology, hydrology, and geochemistry of the aquifer system and the ground-water-flow regime on a regional scale.

All the regional studies use computer models to simulate flow within an aquifer system, but, unlike many studies that stress the importance of the predictive capabilities of calibrated

ground-water-flow models, the RASA studies simulate primarily for analysis (Bennett, 1979). Thus, computer models are used to test hypotheses to increase knowledge of ground-water-flow systems. With this approach, construction of a ground-water-flow model of the Coastal Plain sediments has begun (Meisler, 1980).

The northern Atlantic Coastal Plain includes about 50,000 mi<sup>2</sup>, extending from the North Carolina-South Carolina border, through Long Island, N.Y. (fig. 1). The external boundary of the fresh-water system is complex, as shown by the shape of the Fall Line onshore and the 10,000 milligrams-per-liter chloride-concentration line offshore (Meisler, 1981). The concentration line is assumed to approximate the seaward limit of the freshwater-flow system. Ten aquifers and nine intervening confining beds in the ground-water system have been delineated. Because of the large area of the project and the presence of numerous confining-bed and aquifer pinchouts in the system, a modified version of the Trescott model was needed. This report presents (1) conceptual changes in the existing flow model required to reduce computer-memory requirements and better simulate confining-bed and aquifer pinchouts, (2) modifications required in the model input, (3) the modified source code, and (4) a cost comparison of the modified and Trescott (1975) versions of the three-dimensional ground-water-flow model.

#### MEMORY REDUCTION

To understand the method of memory reduction, knowledge of the nodal numbering or indexing scheme used in the unmodified Trescott model is required. The finite-difference approach requires subdividing of the aquifer system into discrete blocks, which are assumed to have uniform hydraulic properties. A point in the center of the finite-difference block is known as the node and is commonly numbered to conveniently locate the position of the node in the discretized system. A finite-difference approximation of the differential equations of three-dimensional ground-water-flow results in a system of algebraic equations. A generalized equation for this system at node i,j,k is:

$$B_{i,j,k} h_{i-1,j,k} + D_{i,j,k} h_{i,j-1,k} + E_{i,j,k} h_{i,j,k} + F_{i,j,k} h_{i,j+1,k} + H_{i,j,k} h_{i+1,j,k} + S_{i,j,k} h_{i,j,k+1} + Z_{i,j,k} h_{i,j,k-1} = \frac{-S'_{i,j,k}}{\Delta t} h_{i,j,k} \quad (1)$$

where  $B_{i,j,k}$ ,  $D_{i,j,k}$ ,  $E_{i,j,k}$ ,  $F_{i,j,k}$ ,  $H_{i,j,k}$ ,  $S_{i,j,k}$ ,  $Z_{i,j,k}$  are

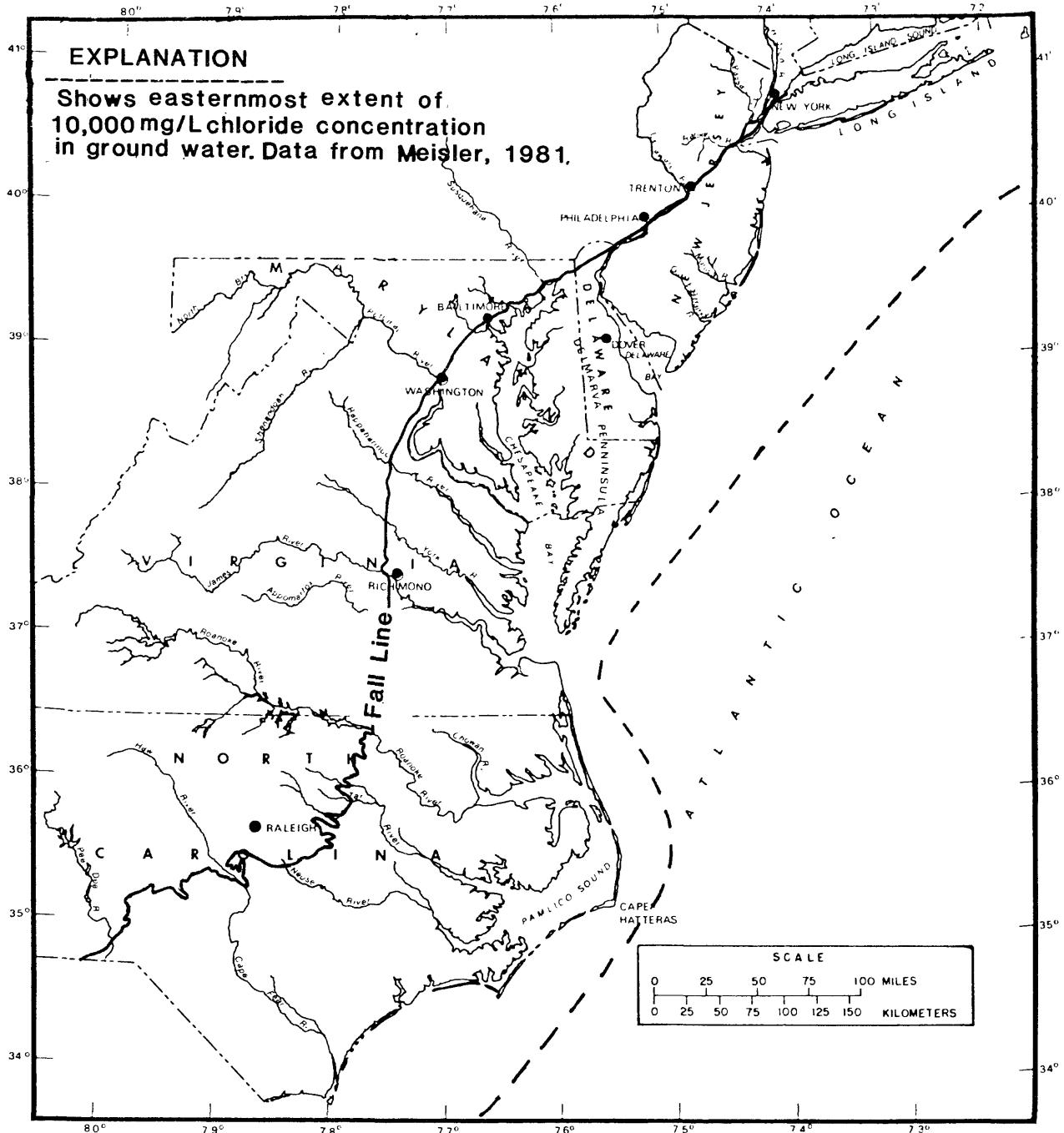


Figure 1.-- Location of northern Atlantic Coastal Plain.

constants, as defined by Trescott (1975). The constants are the harmonic means of the appropriate hydraulic parameters for adjacent blocks. These coefficients provide the correct hydraulic connection between adjacent nodes.

S' is storage coefficient;

$\Delta t$  is length of time step;

$\hat{h}_{i,j,k}$  is the hydraulic head (known) at time  $t$ ;

$h_{i,j,k}$  is the hydraulic head (unknown) at time  $t + \Delta t$ ;

i is the node number in the y-direction;

j is the node number in the x-direction;

k is the node number in the z-direction.

The location of these parameters in a finite-difference mesh is shown on the left of figure 2. In this example, there are a total of 27 finite-difference blocks, 3 blocks in the x-direction, 3 in the y-direction, and 3 in the z-direction. The blocks can be indexed by the i,j,k notation of Equation 1. The indices of each block are shown within the block of figure 2. However, for computation, a more efficient notation is numbering the nodes sequentially (from 1 to 27 in this example), as shown by Trescott, (1975, p. II-5). The i,j,k indices are converted to one index, N, by the following equation:

$$N = i + (j-1)*IO + (k-1)*IO*JO \quad (2)$$

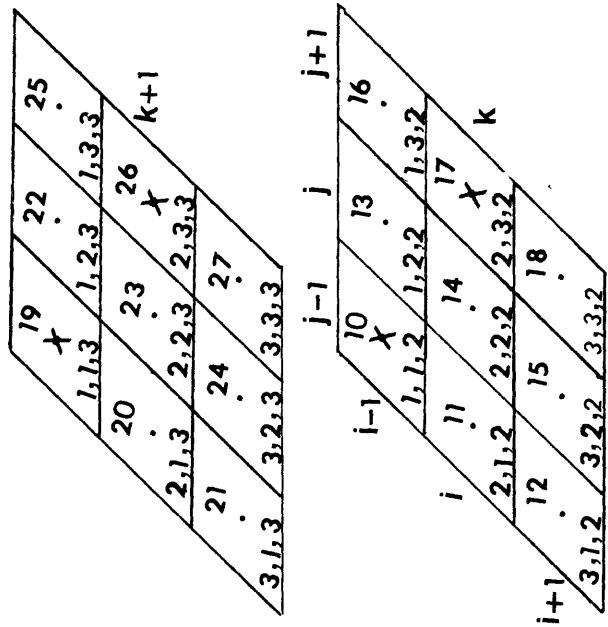
where IO is the total number of blocks in the y-direction and JO is the total number of blocks in the x-direction.

In a finite-difference model, the model grid is always rectangular, whereas the modeled area is generally not. Hence, some nodes in the grid are usually outside the modeled area. Nodes outside the modeled area are designated as no-flow nodes. In the Trescott model, all nodes are numbered, and data are stored at all nodes, including the nodes outside the modeled area. Memory can, therefore, be saved by eliminating the storage of data at these nodes. Because regional models of the scope of the northern Atlantic Coastal Plain need many nodes to represent the aquifer system, wasted computer-memory is not only costly but may affect the choice of the grid spacing used in the model and the computer on which the model will be run. That is, computers have a finite-memory limit, which restricts the size of model that can be run, regardless of cost.

In an aquifer system that contains a large percentage of nodes outside the modeled area, a more efficient method of indexing is numbering only the nodes within the modeled area. These nodes will be referred to as active nodes in the remainder of the report, as opposed to inactive nodes, which are defined in this report as nodes outside the modeled area. The modified model



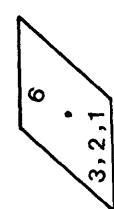
Tresscott Model



5

EXPLANATION

Model block showing block number  $N_{\text{B}}$  and indices  $i=3$ ,  $j=2$  and  $k=1$ .  $N=i+(j-1)IO+(k-1)IO_X$ . In this example  $IO$  is 3 and  $JO$  is 3. The inactive nodes are denoted with an  $X$ .



Model block showing block number  $N = 3$   
 indices  $ij = 6$  and  $k = 1$ .  $N = ij + (k-1)$  IOJO.  
 In this example IOJO is 7, the total number  
 of active nodes in a layer.

**Figure 2.**--Numbering scheme for finite-difference mesh used by the forecast (S).

uses this indexing scheme and takes advantage of the accompanying reduction in computer-memory. The indexing scheme used in the modified model is shown on the right of figure 2. In the example, there are two inactive nodes in each layer, thus, numbering only the active nodes results in a total of 21 finite-difference blocks. In comparison, the standard Trescott indexing scheme results in 27 blocks requiring storage. The  $i,j$  indices are not directly used in this indexing scheme; however, a variable  $ij$  is used. In any layer, the number of a node is  $ij$ , where  $ij$  refers only to the active nodes in the layer. The nodal index  $N$  is given by the following equation:

$$N = ij + (k-1)*IOJO \quad (3)$$

where;  $ij$  has a range from 1 to  $IOJO$ ,

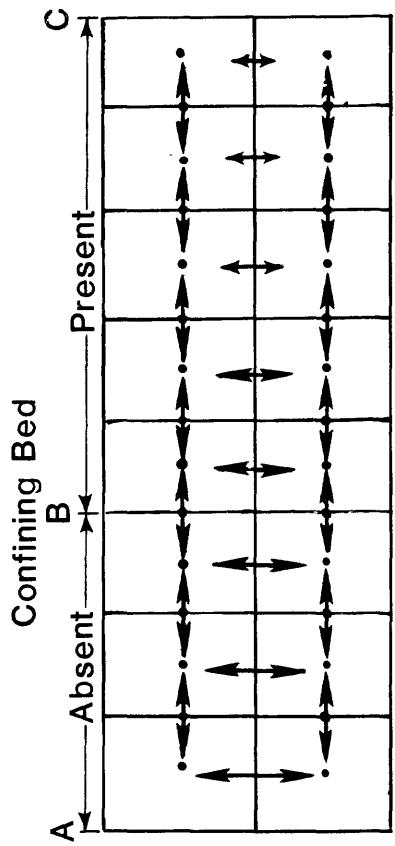
$k$  is the number of the layer of interest, and  $IOJO$  is the number of active nodes in any layer.

#### SIMULATING CONFINING BED AND AQUIFER PINCHOUTS

The Trescott model code uses the transmissivity ( $T$ ) matrix as a flag for determining if a node is included ( $T>0$ ) or excluded ( $T=0$ ) in the computational scheme. The solution process sweeps through all the nodes sequentially, and a check of the value of transmissivity determines the disposition of the node in the model. However, in certain hydrologic situations, the transmissivity matrix alone is not adequate to define the disposition of the node in the model.

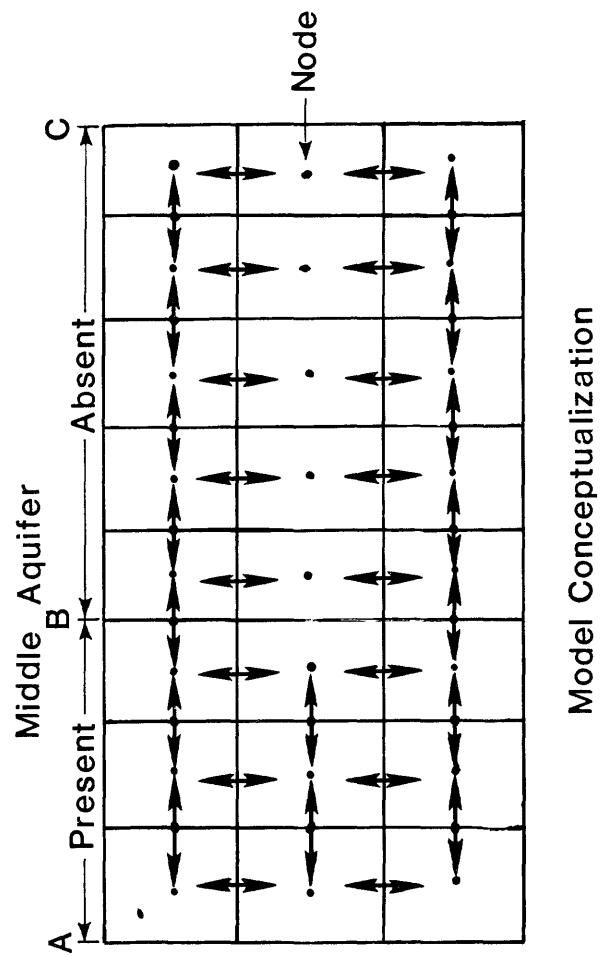
Figure 3 shows typical cross sections, with pinchouts of aquifers and confining beds, and the model conceptualizations of the system in a discretized domain. In case one, the confining bed pinches out. In both the Trescott and modified models, the lateral flow in the aquifers is controlled by the transmissivities; the vertical connection between the aquifers is controlled by the appropriate vertical leakance ( $K_v/b$ ). In areas where the confining bed is absent, the vertical hydraulic properties of the aquifer are used in the model, whereas the vertical properties of the confining bed are used where it is present. Typically, the vertical leakance of the aquifers is much larger than that of the confining beds.

An example of an aquifer pinching out between two adjacent confining beds is shown in case 2. The conceptualization used in the modified model is such that lateral flow in the aquifer is controlled by the transmissivity. Where the aquifer is absent, the transmissivity is specified as zero (no lateral flow), and all flow is in the vertical direction, controlled by the vertical hydraulic properties of the adjacent confining beds. In contrast,



Case 1—Confining Bed Pinchout

Arrows show vertical and horizontal connection between nodes. Length of arrows represents amount of hydraulic connection.



Case 2—Aquifer Pinchout  
Physical Configuration  
Model Conceptualization

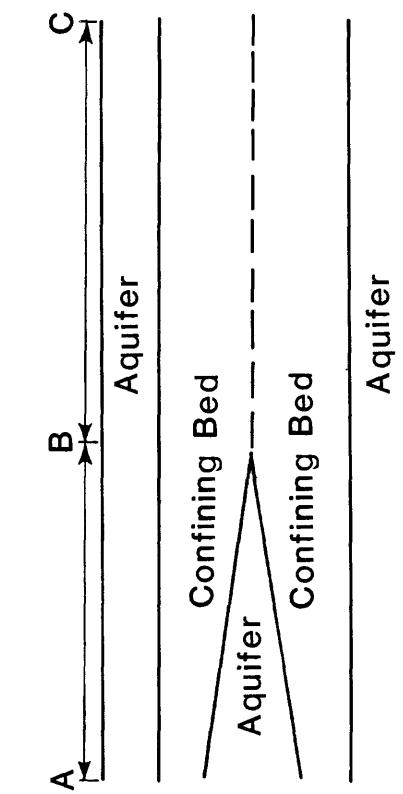
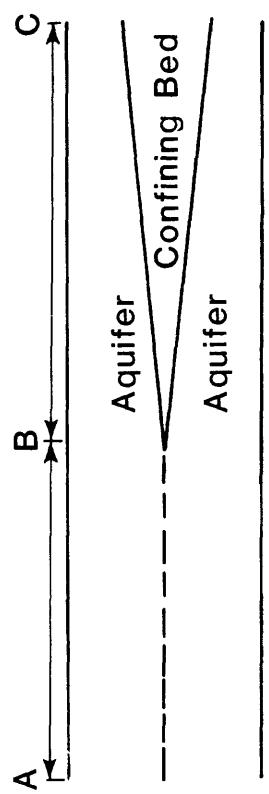


Figure 3.—Generalized cross sections and conceptualizations of aquifer and confining-bed pinchouts.

the logic used in the Trescott model requires that an artificial value of transmissivity greater than zero be specified in the areas where the aquifer is absent. If the transmissivity specified is very small, the lateral flow in this area becomes negligible, and the appropriate hydraulic connections are simulated. Hence, to simulate aquifer pinchouts accurately with the Trescott model artificial values of transmissivity are required to insure that these areas are included in the solution scheme. These artificial values cause unnecessary computational effort (lateral flow), and model results may be sensitive to the value of transmissivity specified.

To model subcrops and pinchouts accurately, the modified model was developed, so that the transmissivity matrix alone is not used to decide whether a node is included in the computational scheme. The modified model checks for vertical hydraulic connections as well as horizontal connections. If both the transmissivity and vertical leakance ( $K_v/b$ , variable TK in the model) are zero, the node is skipped in the computational scheme. However, if either the transmissivity or TK are nonzero, the node is retained.

Besides the changes in program logic, the user must calculate and enter appropriate transmissivity and TK values to match the modified conceptualization of the system. However, TK and transmissivity values are read into the model exactly as before.

In areas where an aquifer is absent between adjacent confining beds, the aquifer transmissivity should be set to zero, and appropriate confining-bed leakance values should be entered to control the vertical flow from one confining bed to another. The head computed for the node represents the head at the contact of adjacent confining beds.

In areas where a confining bed pinches out, the vertical flow between adjacent aquifers is controlled by the vertical hydraulic properties of the aquifers. The harmonic mean of the TK values of adjacent aquifers is commonly used as the effective TK between them. The harmonic mean is  $2*TK1*TK2/(TK1 + TK2)$  where TK1 and TK2 are  $K_v/b$  of the adjacent aquifers.

#### INPUT REQUIREMENTS

Model-program modifications required to implement the memory reduction occur in every subroutine. Also, minor modifications to the input requirements of the program were necessary. Input instructions are given in Appendix I, and the modified model program is listed in Appendix II. The model input is essentially identical to the standard flow model, making the model easily compatible with available input data decks with minor additions. Data additions include (1) defining the starting and ending column of active or modeled nodes in each row in the model and (2) the

total number of active nodes in a composite layer. For example, if layer 1 in the model has active nodes in columns 5 through 15 and layer 2 has active nodes in columns 8 through 25, the composite layer would have active nodes in columns 5 through 25. The starting and ending position of active or modeled nodes in each row in the model are input as two additional data arrays in the model. The size of each array is 10, the number of rows in the model. Within the modeled part of the finite-difference mesh, nodes representing areas where an aquifer is absent can still be indicated by zero transmissivity.

Even though data at unmodeled nodes are not stored, model data values are input for all nodes in the same format as in the standard flow model. The modified program uses the information that defines unused nodes to skip over unneeded data on input. Use of the same input structure makes conversion simpler than if a new input structure were used. On output, printout positions are maintained for all nodes, but, at unused nodes, values are printed as blanks. This makes it readily apparent which nodes are unused.

The program keeps track of the physical location of the adjacent nodes in relation to the node of interest during computation. This assures that the correct lateral hydraulic connections (coefficients TR and TC) are computed in SUBROUTINE COEF, and the solution algorithm is properly formulated in SUBROUTINE SOLVE.

To insure compatibility with the standard Trescott model, the example problem in Trescott (1975, Appendix IV) was simulated with the modified model. In this simulation, the storage for the zero-transmissivity nodes surrounding the active nodes, with the exception of a single dummy node in both rows 1 and 20, was eliminated from the Y-vector (the variable that contains storage for all array data). There were 74 inactive nodes per layer in the modified model simulation out of a total of 400 nodes per layer. The modified model reproduced exactly the results presented in Trescott (1975) for the sample problem.

#### COMPARISON OF MEMORY REQUIREMENTS AND COST FOR SIMULATION OF THE REGIONAL AQUIFER SYSTEM

The preliminary finite-difference grid used to discretize the northern Atlantic Coastal Plain aquifer system is shown in figure 4. The grid consists of 85 rows, 32 columns, and 10 layers, totalling 27,200 nodes. Because of the complex external geometry, not all of these nodes are actually needed to represent the aquifer system. The modeled or active part of the grid is bounded on the west by the Fall Line (Brown, 1972) and on the east by the seaward limit of the 10,000 mg/L isochlor (Meisler, 1981).

## COLUMNS

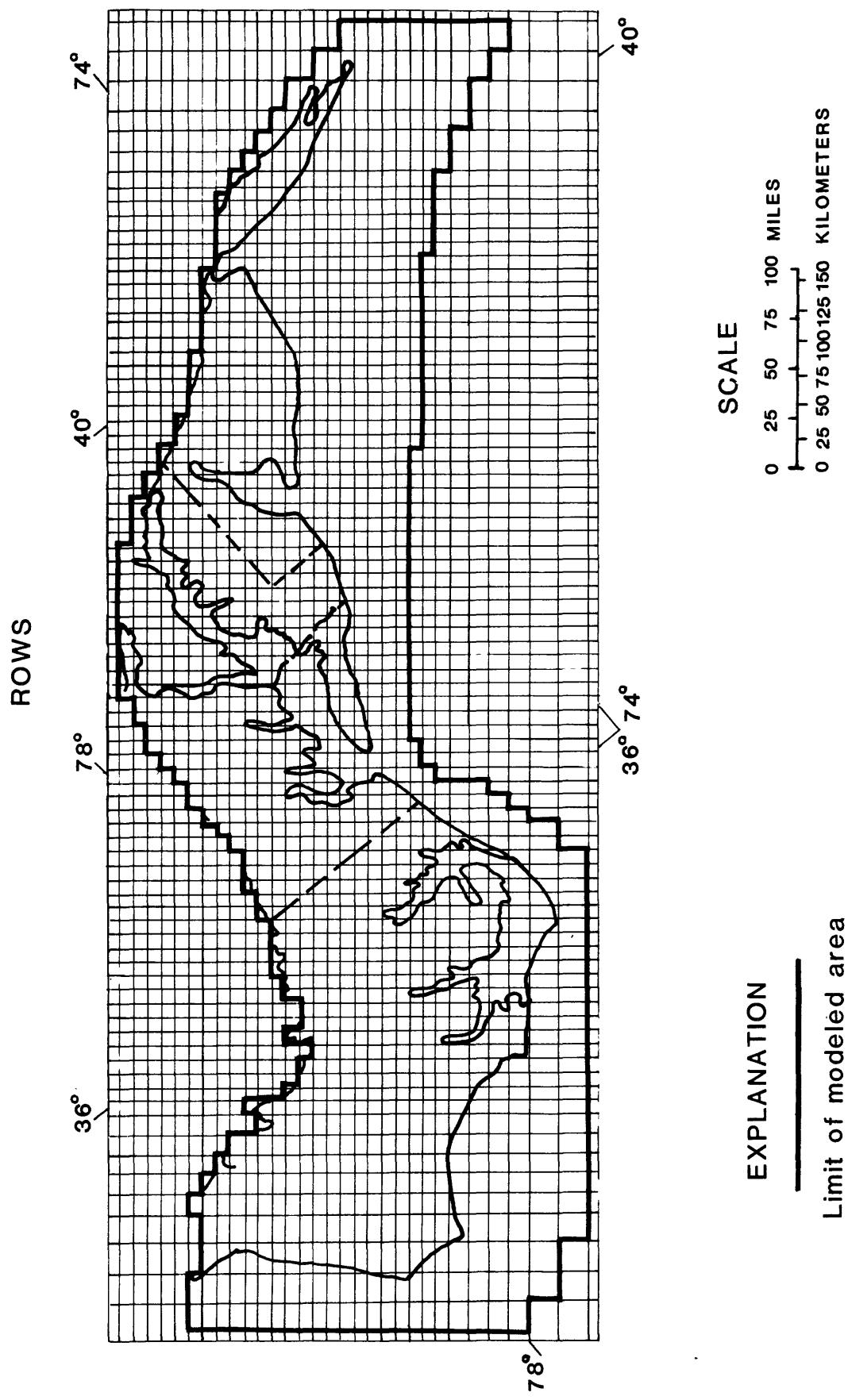


Figure 4.--Finite-difference grid used to discretize the northern Atlantic Coastal Plain aquifer system.

The outline on the grid defines the maximum number of active nodes for a composite of all the layers in the aquifer system. Although some inactive nodes are present within the outline, their number is small. The outline surrounds 15,640 nodes, or 57.5 percent of the 27,200 in the entire grid. The FORTRAN code was modified to store data only for nodes contained within the area defined by the outline (fig. 4).

Simulations of the northern Atlantic Coastal Plain aquifer system model were used to compare costs of the modified model and the Trescott model. Table 1 shows the results. As previously stated, 42.5 percent of the total 27,200 nodes are inactive in this model (simulation 1). The length of the Y-vector required for the simulation is smaller with the modified version. In general, savings were largely on memory costs; savings were also appreciable, however, on cost of execution time. The modified model executes faster because fewer time-consuming disk input and output operations associated with a virtual memory system are required with the smaller memory space. In general, the overall savings increase with an increasing ratio of inactive to total number of nodes. This is demonstrated by simulation 2 (table 1), which is identical to simulation 1 except that only the inactive nodes in the artificial zero transmissivity border (columns 1 and 32, rows 1 and 85) have been removed from storage. With this change, 91.6 percent of the total 27,200 nodes require storage space in the Y-vector.

As a further test of the modified model, the Coastal Plain aquifers of New Jersey were simulated both by the modified and the standard Trescott versions. The modified model duplicated the results of the Trescott model. A cost comparison of these simulations is given in table 1, simulation 3. The total number of nodes in these simulations is 14,790, of which 10 percent are inactive. Because, overall, there are fewer total nodes in this simulation than in the regional model, the memory savings are less. As expected, the overall savings, in comparison to simulation 1, are less.

Table 1.--Cost savings realized by modified model  
for sample problems.

Simula- tion	Number of total nodes	Number of nodes requiring storage	Percent- age of total	Percentage cost savings realized by use of modified rather than Trescott model for sample problems		
				Overall	CPU	Core
1	27,200	15,640	57.5	30	18	45
2	27,200	24,920	91.6	14	3.5	25.2
3	14,790	13,250	90.0	8	4.4	13

## CONCLUSIONS

The modified version of the ground-water-flow model duplicates exactly the results of the standard Trescott model for the sample problem and other test problems. The modified model permits the simulation of confining-bed and aquifer pinchouts without the use of artificial hydraulic parameters and eliminates wasted computer memory. Simulating a large number of nodes, a large percentage of which are inactive, is less expensive if the modified model is used. The savings over the Trescott model are problem dependent, that is, the savings are related to the number of inactive nodes in the simulation. Numerical experiments with the steady-state simulation of the northern Atlantic Coastal Plain regional aquifer study showed overall savings of 30 percent. In general, appreciable savings in simulating regional aquifer systems with complex external boundaries can be realized by the modified model.

## REFERENCES

- Bennett, G. D., 1979, Regional ground-water systems analyses: Water Spectrum, v. 11, no. 4, p. 36-42.
- Brown, P. M., Miller, J. A., and Swain, F. M., 1972, Structural and stratigraphic framework, and spatial distribution of permeability of the Atlantic Coastal Plain, North Carolina to New York: U.S. Geological Survey Professional Paper 796, 79 p.

REFERENCES--Continued

- Meisler, Harold, 1980, Plan of study for the Northern Atlantic Coastal Plain regional aquifer system analysis: U.S. Geological Survey Water-Resources Investigations 80-16, 27 p.
- Meisler, Harold, 1981, Preliminary delineation of salty ground water in the northern Atlantic Coastal Plain: U.S. Geological Survey Open-File Report 81-71, 34 p.
- Posson, D. R., Hearne, G. A., Tracy, J. V., and Frenzel, P. F., 1980, A computer program for simulating geohydrologic systems in three dimensions: U.S. Geological Survey Open-File Report 80-421, 795 p.
- Trescott, P. C., 1975, Documentation of finite-difference model for simulation of three-dimensional ground-water flow: U.S. Geological Survey Open-File Report 75-438, 103 p.

## APPENDIX 1

DATA DECK INSTRUCTIONS

The data deck instructions have been made as compatible as possible with the Trescott version of the flow model. This allows existing data decks to be used with the modified model with minimal amount of recoding. The modified model also includes the transient leakage option (ENTRY CLAY) described by Posson, Hearne, Tracy and Frenzel (1980) and a modification to allow a user-specified maximum iteration parameter (WMAX) in ENTRY ITER. Because contouring packages for pen plotters are generally available, the line printer plotting routine (SUBROUTINE PRINTAI) was eliminated. The instructions that follow are adapted from Open-File Report 75-438.

Group I: Title, Simulation Options and Problem Dimensions

This group of cards, which are read by the main program, contains data required to dimension the model. To specify an option on card 4 punch the characters underlined in the definition. For an option not used, that section of card 4 can be left blank.

Note: Default typing of variables applies for all data input.

<u>CARD</u>	<u>COLUMNS</u>	<u>FORMAT</u>	<u>VARIABLE</u>	<u>DEFINITION</u>
1	1-80	20A4	HEADING	Any title the user wishes to print on one line at the start of output
2	1-52	13A4	HEADING	
3	1-10	I10	IO	Number of rows
	11-20	I10	JO	Number of columns
	21-30	I10	KO	Number of layers
	31-40	I10	ITMAX	Maximum number of iterations per time step
	41-50	I10	NCH	Number of constant head nodes
	51-60	I10	MODE	Number of terms used in transient leakage code
	61-70	I10	IOJO	Number of modeled nodes per layer

## APPENDIX 1

DATA DECK INSTRUCTIONS--Continued

NOTE: IOJO is the total number of modeled nodes for a composite layer consisting of a combination of all layers considered together. In other words, the position of JDIML1 may be determined by the external geometry of Layer 1, whereas the position of JDIML2 may be determined by the external geometry of Layer 2. The variable IOJO would be the number of modeled nodes in this composite layer.

<u>CARD</u>	<u>COLUMNS</u>	<u>FORMAT</u>	<u>VARIABLE</u>	<u>DEFINITION</u>
4	1-4	A4	IDRAW	<u>DRAW</u> to print drawdown
	6-9	A4	IHEAD	<u>HEAD</u> to print hydraulic head
	11-14	A4	IFLOW	<u>MASS</u> to compute a mass balance
	16-18	A3	IDK1	<u>DK1</u> to read initial head, elapsed time, and mass balance parameters on unit 4 (disk)
	21-23	A3	IDK2	<u>DK2</u> to write computed head, elapsed time, and mass balance parameters on unit 4 (disk)
	26-29	A4	IWATER	<u>WATE</u> if the upper hydrologic unit is unconfined
	31-34	A4	IQRE	<u>RECH</u> for a constant recharge that may be a function of space
	36-39	A4	IPU1	<u>PUN1</u> to read initial head, elapsed time, and mass balance parameters from cards
	41-44	A4	IPU2	<u>PUN2</u> to punch computed head, elapsed time, and mass balance parameters on cards
	46-49	A4	ITK	<u>ITKR</u> to read the value of $TK(I, J, K)$ for simulations in which confining layers are not represented by layers of nodes. $TK(I, J, K) = Kv/b$ .
	51-54	A4	IEQN	<u>EQN3</u> to solve Equation 3

## APPENDIX 1

DATA DECK INSTRUCTIONS--Continued

<u>CARD</u>	<u>COLUMNS</u>	<u>FORMAT</u>	<u>VARIABLE</u>	<u>DEFINITION</u>
	56-59	A4	ITL	ITLR to read values of Rate (I,J,K-1) M(I,J,K-1),SS(K-1)

Note: For continuation of a simulation, if transient leakage option (ITLR) is specified variables RM, XI, and DELT are written, punched or read on disk or cards dependent on options specified (DK1, DK2, PUN1, PUN2).

Group IA: Modeled nodes array

<u>ARRAY</u>	<u>COLUMNS</u>	<u>FORMAT</u>	<u>VARIABLE</u>	<u>DEFINITION</u>
1	1-80	8I10	JDIML1(IO)	Location of first modeled node in each row in x-direction
2	1-80	8I10	JDIML2(IO)	Location of last modeled node in each row in x-direction

NOTE: The first modeled node may not appear in column 1 and may not appear after column 40.

Group II: Scalar parameters

The parameters required in every problem are underlined. The other parameters are required as noted; when not required, their location on the card can be left blank. The G format is used to read E, F and I format data. Minimize mistakes by always right-justifying data in the field. If F format data do not contain significant figures to the right of the decimal point, the decimal point can be omitted.

<u>CARD</u>	<u>COLUMNS</u>	<u>FORMAT</u>	<u>VARIABLE</u>	<u>DEFINITION</u>
1	1-10	G10.0	<u>NPER</u>	Number of pumping periods for the simulation
	11-20	G10.0	<u>KTH</u>	Number of time steps between printouts

NOTE: To print only the results for the final time step in a pumping period, make KTH greater than the expected number of time steps. The program always prints the results for the final time step.

21-30      G10.0      ERR      Error criteria for closure (L)

## APPENDIX 1

DATA DECK INSTRUCTIONS--Continued

NOTE: When the head change at all nodes on subsequent iterations is less than this value (for example, 0.01 foot), the program has converged to a solution for the time step.

<u>CARD</u>	<u>COLUMNS</u>	<u>FORMAT</u>	<u>VARIABLE</u>	<u>DEFINITION</u>
	31-40	G10.0	<u>LENGTH</u>	Number of iteration parameters
2	1-20	G20.10	SUM	
	21-40	G20.10	SUMP	
	41-60	G20.10	PUMPT	Parameters in which elapsed time and cumulative volumes for mass balance are stored.
	61-80	G20.10	CFLUXT	For the start of a simulation insert three blank cards. <u>For continuation</u> of a previous run using cards as input, replace the three blank cards with the first three cards of punched output from the previous run.
3	1-20	G20.10	QRET	Using data from disk for input, leave the three blank cards in the data deck.
	21-40	G20.10	CHST	
	41-60	G20.10	CHDT	
	61-80	G20.10	FLUXT	
4	1-20	G20.10	STORT	
4	21-40	G20.10	ETFLXT	
	41-60	G20.10	FLXNT	

Group III: Array Data

Each of the following data sets (except data set 1) consists of a parameter card and, if the data set contains variable data, a set of data cards for each layer in the model. Each parameter card contains at least five variables. The final card in Group III is not a part of an array data set but rather a single card defining the maximum iteration parameter.

## APPENDIX 1

DATA DECK INSTRUCTIONS--Continued

<u>CARD</u>	<u>COLUMNS</u>	<u>FORMAT</u>	<u>VARIABLE</u>	<u>DEFINITION</u>
Every Parameter Card	1-10	G10.0	FAC	If IVAR = 0, FAC is the value assigned to every element of the matrix for this layer.
				If IVAR = 1, FAC is the multiplication factor for the following set of data cards for this layer.
	11-20	G10.0	IVAR	= 0 if no data cards are to be read for this layer. = 1 if data cards for this layer follow.
	21-30	G10.0	IPRN	=0 if input data for this layer are to be printed. = 1 if input data for the layer are <u>not</u> to be printed.

Transmissivity parameters Cards also have these Variables	31-40	G10.0	<u>FACT(K,1)</u>	multiplication factor for transmissivity in x-direction
	41-50	G10.0	<u>FACT(K,2)</u>	multiplication factor for transmissivity in the y-direction
	51-60	G10.0	<u>FACT(K,3)</u>	multiplication factor for hydraulic conductivity in the z-direction. (Not used when confining bed nodes are eliminated and TK values are read)

## APPENDIX 1

DATA DECK INSTRUCTIONS--Continued

<u>CARD</u>	<u>COLUMNS</u>	<u>FORMAT</u>	<u>VARIABLE</u>	<u>DEFINITION</u>
Every Parameter Card	61-70	G10.0	IRECS	= 0 if the matrix is being read from cards or if each element is being set equal to FAC. = 1 if the matrix is to be read from disk (unit 2)
	71-80	G10.0	IRECD	= 0 if the matrix is <u>not</u> to be stored on disk. = 1 if the matrix being read from cards or set equal to FAC <u>is</u> to be stored on disk (unit 2) for later retrieval.

When data cards are included, start each row on a new card. To prepare a set of data cards for an array that is a function of space, the general procedure is to overlay the finite-difference grid on a contoured map of the parameter and record the average value of the parameter for each finite-difference block on coding forms according to the appropriate format. In general, record only significant digits and no decimal points (except for data set 2); use the multiplication factor to convert the data to their appropriate values. For example, if DELX ranges from 1,000 to 15,000 feet, coded values should range from 1-15; the multiplication factor (FAC) would be 1,000.

<u>DATA SET</u>	<u>COLUMNS</u>	<u>FORMAT</u>	<u>VARIABLE</u>	<u>DEFINITION</u>
1	1-80	8F10.4	PHI(IJ,K)	Head values for continuation of a previous run (L)

NOTE: For a new simulation this data set is omitted. Do not include a parameter card with this data set.

2	1-80	8F10.4	STRT(IJ,K)	Starting head matrix (L)
3	1-80	20F4.0	S (IJ,K)	Storage coefficient (dimensionless)

NOTE: This matrix is also used to locate constant head boundaries by coding a negative number at constant head nodes.

## APPENDIX 1

DATA DECK INSTRUCTIONS--Continued

<u>DATA SET</u>	<u>COLUMNS</u>	<u>FORMAT</u>	<u>VARIABLE</u>	<u>DEFINITION</u>
4	1-80	20F4.0	T(IJ,K)	Transmissivity ( $L^2/t$ )
NOTE: 1) See the previous page for the additional requirements on the parameter cards for this data set.				
	2)		If the upper modeled layer is unconfined and PERM and BOTTOM are to be read for this layer, insert a parameter card <u>for this layer</u> with only the values for FACT on it.	
	3)		Zero values may be used to represent aquifer pinchouts. Finite values of TK will allow flow directly from confining bed to confining bed in these areas.	
5	1-80	8F10.4	TK(IJ,K)	= $K_v/b$ leakance (/T).
NOTE: This data set is read for both steady leakage(ITKR) and transient leakage(ITLR) options. If the Transient Leakge Option is specified, zero values must be specified for TK. The number of layers of TK values is KO-1.				
6	1-80	20F4.0	PERM(IJ)	Hydraulic conductivity (L/T) (see note 1 for data set 4)
7	1-80	20F4.0	BOTTOM(IJ)	Elevation of bottom of water-table unit (L)
NOTE: Data set 6 and 7 are required only for simulating unconfined conditions in the upper hydrologic unit.				
8	1-80	20F 4.0	QRE(IJ)	Recharge rate (L/T)
NOTE: Omit if not used				
9	1-80	20F 4.0	RATE(IJ,K)	Vertical hydraulic conductivity of confining bed ( $K'$ ) (L/T)
10	1-80	20F 4.0	ZCB(IJ,K)	Thickness of confining bed (L)

## APPENDIX 1

DATA DECK INSTRUCTIONS--Continued

<u>DATA SET</u>	<u>COLUMNS</u>	<u>FORMAT</u>	<u>VARIABLE</u>	<u>DEFINITION</u>
-----------------	----------------	---------------	-----------------	-------------------

11	1-80	8G10.0	SS(K)	Specific storage of confining bed (/L)
----	------	--------	-------	--

NOTE: Data sets 9, 10, and 11 are required for simulating transient leakage from confining beds. If steady leakage is required for a particular confining bed in the simulation, specific storage for the particular confining bed can be specified as zero, or rate can be specified as zero and a finite value of TK for the confining bed can be used.

---

12	1-80	8G10.0	DELX(J)	Grid spacing in x direction (L)
----	------	--------	---------	---------------------------------

13	1-80	8G10.0	DELY(I)	Grid spacing in y direction (L)
----	------	--------	---------	---------------------------------

14	1-80	8G10.0	DELZ(K)	Grid spacing in z direction (L)
----	------	--------	---------	---------------------------------

1-10	F10.7	WMAX	Maximum iteration parameter (optimum value can be determined by trial and error for each problem, 0.99863 is a good first guess)
------	-------	------	--

Group IV: Parameters that change with the pumping period

The program has two options for the simulation period:

1. To simulate a given number of time steps, set TMAX to a value larger than the expected simulation period. The program will use NUMT, CDLT, and DELT as coded.
2. To simulate a given pumping period, set NUMT larger than the number required for the simulation period (for example, 100). The program will compute the exact DELT (which will be < DELT coded) and NUMT to arrive exactly at TMAX on the last time step.

## APPENDIX 1

DATA DECK INSTRUCTIONS--Continued

<u>CARD</u>	<u>COLUMNS</u>	<u>FORMAT</u>	<u>VARIABLE</u>	<u>DEFINITION</u>
1	1-10	G10.0	KP	Number of the pumping period
	11-20	G10.0	KPM1	Number of the previous pumping period
NOTE: KPM1 is currently not used				
21-30	G10.0	NWEL		Number of wells for this pumping period
31-40	G10.0	TMAX		Number of days in this pumping period
41-50	G10.0	NUMT		Number of time steps
51-60	G10.0	CDLT		Multiplying factor for DELT
NOTE: 1.5 is commonly used				
61-70	G10.0	DELT		Initial time step in <u>hours</u>

if NWEL = 0 the following set of cards is omitted

DATA SET 1				(NWEL cards)
<u>CARD</u>	<u>COLUMNS</u>	<u>FORMAT</u>	<u>VARIABLE</u>	<u>DEFINITION</u>
	1-10	G10.0	K	Layer in which well is located
	11-20	G10.0	I	Row location of well
	21-30	G10.0	J	Column location of well
	31-40	G10.0	WELL(IJ,K)	Pumping rate ( $L^3/t$ ), negative for a pumping well.

For each additional pumping period, another set of group IV cards is required (that is, NPER sets of group IV cards are required).

APPENDIX 2  
LISTING OF SOURCE CODE

C MODIFIED FLOW MODEL TO REDUCE CORE STORAGE, INCLUDES TRANSIENT LEAKAGE  
C JUNE 1982

---

C FINITE-DIFFERENCE MODEL FOR SIMULATION OF GROUND-WATER FLOW IN  
C THREE DIMENSIONS, SEPTEMBER, 1975 BY P.C. TRECOTT, U. S. G. S.  
C WITH CONTRIBUTIONS TO MAIN, DATA1 AND SOLVE BY S.P. LARSON  
C CHANGES TO REDUCE Y-VECTOR LENGTH BY P.P. LEAHY

---

C SPECIFICATIONS:

REAL \*8YSTR 00000010

DIMENSION Y(20000), L(34), HEADNG(33), NAME(54), DUM(3)  
1, LL(2), JD(100) 00000020

EQUIVALENCE (YSTR,Y(1)) 00000030

COMMON /INTEGR/ IO,JO,KO,I1,J1,K1,NPER,KTH,ITMAX,LENGTH,KP,NWEL,  
1NUMT,IFINAL,IT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCH,  
2IDK1,IDK2,IWATER,IQRE,IPJP,IQJQ,IKJK,K5,IPU1,IPU2,ITK,IEQN,MODE  
. ,ITL,IOJO,K6,K7,ILJL  
COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,QR,DELM1 00000050  
COMMON /SARRAY/ ICHK(13) 00000060

DATA NAME/2\*4H ,4H S,4HTART,4HING ,4HHEAD,4H ,4H STO,4HRAG00000070  
1E,4H COE,4HFFIC,4HIENT,2\*4H ,4H TR,4HANSM,4HISSI,4HVITY,5\*4H  
2 ,4H TK,4H HY,4HDRAU,4HLIC ,4HCOND,4HUCTI,4HVITY,2\*4H ,4HBOT  
3T,4HOM E,4HLEVA,4HTION,2\*4H ,4H R,4HECHA,4HRGE ,4HRATE,4H A  
4,4HQUIT,4HARD ,4HCOND,4HUCTI,4HVITY,4H ,4H AQ,4HUITA,4HRD T,4H  
5HICK,4HNESS/

DEFINE FILE 2(8,1520,U,KKK) 00000080  
ILENGY=20000 00000090  
JLENG=100 00000100

.....  
C ---READ TITLE, PROGRAM SIZE AND OPTIONS---

READ (5,200) HEADNG 00000110

WRITE (6,190) HEADNG 00000120

READ (5,160) IO,JO,KO,ITMAX,NCH,MODE,IOJO 00000130

WRITE (6,180) IO,JO,KO,ITMAX,NCH,MODE,IOJO 00000140

READ (5,210) IDRAW,IHEAD,IFLO,IK1,IK2,IWATER,IQRE,IPU1,IPU2,ITK  
1,IEQN,ITL 00000150

WRITE (6,220) IDRAW,IHEAD,IFLO,IK1,IK2,IWATER,IQRE,IPU1,IPU2,ITK  
1,IEQN,ITL 00000160

IERR=0 00000170

C ---COMPUTE DIMENSIONS FOR ARRAYS---

J1=JO-1 00000180

I1=IO-1 00000190

K1=KO-1 00000200

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

I2=I0-2                                00000210
J2=J0-2                                00000220
K2=K0-2                                00000230
IMAX=MAX0(I0,J0)                        00000240
NCD=MAX0(1,NCH)                         00000250
ITMX1=ITMAX+1                           00000260
ISIZ=IOJO*K0                            00000270
IK1=IOJO                                00000280
IK2=MAX0(IK1*K1,1)                      00000290
ISUM=2*ISIZ+1                           00000300
L(1)=1                                  00000310
DO 30 I=2,14                             00000320
IF (I.NE.8) GO TO 20                     00000330
L(8)=ISUM                               00000340
ISUM=ISUM+IK2                           00000350
IF (IK2.EQ.1) GO TO 10                   00000360
IKJK=IK1                                00000370
K5=K1                                   00000380
GO TO 30                                00000390
10 IKJK=1                                00000400
K5=1                                     00000410
GO TO 30                                00000420
20 L(I)=ISUM                            00000430
ISUM=ISUM+ISIZ                          00000440
30 CONTINUE                               00000450
L(15)=ISUM                            00000460
ISUM=ISUM+J0                            00000470
L(16)=ISUM                            00000480
ISUM=ISUM+IO                            00000490
L(17)=ISUM                            00000500
ISUM=ISUM+KO                            00000510
L(18)=ISUM                            00000520
ISUM=ISUM+IMAX                          00000530
L(19)=ISUM                            00000540
ISUM=ISUM+K0*3                           00000550
L(20)=ISUM                            00000560
ISUM=ISUM+ITMX1                          00000570
L(21)=ISUM                            00000580
ISUM=ISUM+3*NCD                          00000590
L(22)=ISUM                            00000600
ISUM=ISUM+NCD                           00000610
L(23)=ISUM                            00000620
IF (IWATER.NE.ICHK(6)) GO TO 40        00000630
ISUM=ISUM+IK1                           00000640
L(24)=ISUM                            00000650
ISUM=ISUM+IK1                           00000660
IPJP=IK1                                00000670
GO TO 50                                00000680
40 ISUM=ISUM+1                           00000690
L(24)=ISUM                            00000700
ISUM=ISUM+1                           00000710

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

IPJP=1          00000720
50 L(25)=ISUM  00000730
  IF (IQRE.NE.ICHK(7)) GO TO 60  00000740
  ISUM=ISUM+IK1  00000750
  IQJQ=IK1  00000760
  GO TO 70  00000770
60 ISUM=ISUM+1  00000780
  IQJQ=1  00000790
70 L(26) = ISUM  00000800
  IF ( ITL.NE.ICHK(12) )   GO TO 75  00000810
  ILJL=IK1  00000820
  K6=K0  00000830
  K7=K1  00000840
  ISUM = ISUM + IK2  00000850
  L(27) = ISUM  00000860
  ISUM = ISUM + IK2  00000870
  L(28) = ISUM  00000880
  ISUM = ISUM + K1  00000890
  L(29) = ISUM  00000900
  ISUM = ISUM + ISIZ  00000910
  L(30) = ISUM  00000920
  ISUM = ISUM + ISIZ  00000930
  L(31) = ISUM  00000940
  ISUM = ISUM + ISIZ  00000950
  L(32) = ISUM  00000960
  ISUM = ISUM + 2*MODE*IK2  00000970
  GO TO 77  00000980
75 ILJL=1  00000990
  MODE=1  00001000
  K6=1  00001010
  K7=1  00001020
  ISUM=ISUM+1  00001030
  L(27)=ISUM  00001040
  ISUM=ISUM+1  00001050
  L(28)=ISUM  00001060
  ISUM=ISUM+1  00001070
  L(29)=ISUM  00001080
  ISUM=ISUM+1  00001090
  L(30)=ISUM  00001100
  ISUM=ISUM+1  00001110
  L(31)=ISUM  00001120
  ISUM=ISUM+1  00001130
  L(32)=ISUM  00001140
  ISUM=ISUM+2  00001150
77 JSUM = IO  00001160
  LL(1)=1  00001170
  LL(2)=JSUM+1  00001180
  JSUM = JSUM + IO  00001190
  WRITE (6,170) ISUM,ILENGY  00001200
  WRITE (6,175) JSUM,JLENG  00001210
  IF (ISUM.GT.ILENGY)   GO TO 155  00001220

```

```

C
C ---PASS INITIAL ADDRESSES OF ARRAYS TO SUBROUTINES---
CALL DATAI(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y(L(5)),Y(L(6)),Y(L(7))00001230
1,Y(L(8)),Y(L(9)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(19)),Y(L(23)),Y(L(
224)),Y(L(25)),Y(L(14)),Y(L(32)),JD(LL(1)),JD(LL(2)) )
CALL STEP(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y(L(5)),Y(L(6)),Y(L(7)),00001240
1Y(L(8)),Y(L(9)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(19)),Y(L(18)),Y(L(2
20)),Y(L(14)),Y(L(32)),JD(LL(1)),JD(LL(2)) )
CALL SOLVE(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y(L(5)),Y(L(6)),Y(L(7))00001250
1,Y(L(8)),Y(L(9)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(19)),Y(L(10)),Y(L(
211)),Y(L(12)),Y(L(13)),Y(L(14)),Y(L(20)),Y(L(25))
3,Y(L(26)),Y(L(27)),Y(L(28)),Y(L(29)),Y(L(30)),Y(L(31)),Y(L(32)),
4JD(LL(1)),JD(LL(2)) )
CALL COEF(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y(L(5)),Y(L(6)),Y(L(7)),00001260
1Y(L(8)),Y(L(9)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(19)),Y(L(23)),Y(L(2
24)),Y(L(25)),Y(L(14)) )
3,Y(L(26)),Y(L(27)),Y(L(28)),Y(L(29)),Y(L(30)),Y(L(31)),Y(L(32)),
4JD(LL(1)),JD(LL(2)) )
CALL CHECKI(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y(L(5)),Y(L(6)),Y(L(7))00001270
1),Y(L(8)),Y(L(9)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(19)),Y(L(21)),Y(L(
2(22)),Y(L(25)),Y(L(14)) )
3,Y(L(26)),Y(L(27)),Y(L(28)),Y(L(29)),Y(L(30)),Y(L(31)),Y(L(32)),
4JD(LL(1)),JD(LL(2)) )

C
C ---START COMPUTATIONS---
*****
C
C ---READ AND WRITE DATA FOR GROUPS II AND III---
CALL DATAIN                                00001280
IRN=1                                         00001290
NIJ=IOJO                                      00001300
DO 80 K=1,K0                                    00001310
LOC=L(2)+(K-1)*NIJ                            00001320
80 CALL ARRAY(Y(LOC),2,1,NAME(1),IRN,DUM,K)   00001330
DO 90 K=1,K0                                    00001340
LOC=L(5)+(K-1)*NIJ                            00001350
90 CALL ARRAY(Y(LOC),1,2,NAME(7),IRN,DUM,K)   00001360
DO 100 K=1,K0                                   00001370
LOC=L(4)+(K-1)*NIJ                            00001380
L1=L(19)+K-1                                  00001390
L2=L(19)+K0+K-1                               00001400
L3=L(19)+2*K0+K-1                            00001410
CALL ARRAY(Y(LOC),1,2,NAME(13),IRN,DUM,K)    00001420
Y(L1)=DUM(1)                                   00001430
Y(L2)=DUM(2)                                   00001440
Y(L3)=DUM(3)                                   00001450
100 WRITE (6,230) K,Y(L1),Y(L2),Y(L3)        00001460
IF (ITK.NE.ICHK(10)) GO TO 120                00001470
DO 110 K=1,K1                                   00001480
LOC=L(8)+(K-1)*NIJ                            00001490
110 CALL ARRAY(Y(LOC),2,3,NAME(19),IRN,DUM,K) 00001500
120 IF (IWATER.NE.ICHK(6)) GO TO 130          00001510
K=K0                                           00001520
CALL ARRAY(Y(L(23)),1,4,NAME(25),IRN,DUM,K)  00001530
CALL ARRAY(Y(L(24)),1,1,NAME(31),IRN,DUM,K)  00001540

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

130 IF (IQRE.EQ.ICHK(7)) CALL ARRAY(Y(L(25)),1,4,NAME(37),IRN,DUM,K) 00001550
    IF (ITL.NE.ICHK(12) ) GO TO 135                                00001560
    DO 131 K=1,K1                                              00001570
    LOC = L(26) + (K-1)*NIJ                                         00001580
131 CALL ARRAY (Y(LOC),1,3,NAME(43),IRN,DUM,K)                      00001590
    DO 132 K=1,K1                                              00001600
    LOC = L(27) + (K-1)*NIJ                                         00001610
132 CALL ARRAY (Y(LOC),1,1,NAME(49),IRN,DUM,K)                      00001620
    READ (5,240) FAC,IVAR,IPRN                                     00001630
    LOC = L(28)                                              00001640
    IF (IVAR.EQ.1) READ (5,240)(Y(LOC+K-1),K=1,K1)                00001650
    DO 133 K=1,K1                                              00001660
    IF (IVAR.NE.1) GO TO 134                                00001670
    Y(LOC+K-1) = Y(LOC+K-1)*FAC                               00001680
    GO TO 133                                              00001690
134 Y(LOC+K-1) = FAC                                         00001700
133 CONTINUE                                           00001710
    IF (IVAR.EQ.1.AND.IPRN.NE.1) WRITE (6,250) (Y(LOC+K-1),K=1,K1) 00001720
    IF (IVAR.EQ.0) WRITE (6,260) FAC                           00001730
135 CONTINUE                                           00001740
    CALL MDAT                                              00001750

C
C   ---COMPUTE TRANSMISSIVITY FOR UNCONFINED LAYER---
C   IF (IWATER.EQ.ICHK(6)) CALL TRANS(1)                         00001760
C
C   ---COMPUTE T COEFFICIENTS---
C   CALL TCOF                                              00001770
C
C   ---COMPUTE ITERATION PARAMETERS---
C   CALL ITER                                              00001780
C
C   ---READ TIME PARAMETERS AND PUMPING DATA FOR A NEW PUMPING PERIOD-
140 CALL NEWPER                                         00001790
C
C     KT=0                                                 00001800
C     IFINAL=0                                            00001810
C
C   ---START NEW TIME STEP COMPUTATIONS---
150 CALL NEWSTP                                         00001820
C
C   ---START NEW ITERATION IF MAXIMUM NO. ITERATIONS NOT EXCEEDED---
C   CALL NEWITA                                         00001830
C
C   ---PRINT OUTPUT AT DESIGNATED TIME STEPS---
C   CALL OUTPUT                                         00001840
C
C   ---LAST TIME STEP IN PUMPING PERIOD μ---
C   IF (IFINAL.NE.1) GO TO 150                            00001850
C
C   ---CHECK FOR NEW PUMPING PERIOD---
C   IF (KP.LT.NPER) GO TO 140                            00001860

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

C      155 STOP                               00001870
C
C      ---FORMATS---
C
C
C      160 FORMAT (8I10)                      00001880
C      170 FORMAT ('0',54X,'WORDS OF VECTOR Y USED =',I7,8X,' OF Y = ',I7) 00001890
C      175 FORMAT ('0',54X,'WORDS OF VECTOR JD USED =',I6,8X,' OF JD = ',I6) 00001900
C      180 FORMAT ('0',62X,'NUMBER OF ROWS =',I5/60X,'NUMBER OF COLUMNS =',I500001910
C          1/61X,'NUMBER OF LAYERS =',I5//39X,'MAXIMUM PERMITTED NUMBER OF ITE
C          2RATIONS =',I5//48X,'NUMBER OF CONSTANT HEAD NODES =',I5//44X,
C          .'NUMBER OF TRANSIENT LEAKAGE MODES =',I5//44X,'NUMBER OF ACTIVE NO
C          .DES PER LAYER =',I5)                 00001910
C      190 FORMAT ('1',33A4)                     00001920
C      200 FORMAT (20A4)                       00001930
C      210 FORMAT (16(A4,1X))                  00001940
C      220 FORMAT ('-SIMULATION OPTIONS: ',12(A4,4X)) 00001950
C      230 FORMAT (1H0,44X,'DIRECTIONAL TRANSMISSIVITY MULTIPLICATION FACTORS00001960
C          1 FOR LAYER',I3/,76X,'X =',1P1G15.7/76X,'Y =',1P1G15.7/76X,'Z =',
C          .1P1G15.7)
C      240 FORMAT (8G10.0)                      00001970
C      250 FORMAT (1H ,46X,'SPECIFIC STORAGE'/47X,40(''')//('0',1P18G15.7)) 00001980
C      260 FORMAT ('0',55X,'ALL SPECIFIC STORAGES = ',1P1G15.7)            00001990
C      270 FORMAT (1H1,46X,'POSITION OF FIRST ACTIVE NODE IN X DIRECTION'/
C          .47X,44(''')//('0',12I10))           00002000
C      280 FORMAT (1H1,46X,'POSITION OF LAST ACTIVE NODE IN X DIRECTION'/
C          .47X,43(''')//('0',12I10))
C      END                                     00002020
C      SUBROUTINE DATAI(PHI,STRT,OLD,T,S,TR,TC,TK,WELL,DELX,DELY,DELZ,FAC00002030
C          1T,PERM,BOTTOM,QRE,XI,RM,JDIML1,JDIML2)
C
C      -----
C      READ AND WRITE DATA
C
C
C      SPECIFICATIONS:
C      REAL *8PHI                                00002040
C          REAL*4 IOFT1(12),INFT1(12)             00002050
C
C      DIMENSION PHI(IOJO,KO), STRT(IOJO,KO), OLD(IOJO,KO), T(IOJO,KO
C          1), S(IOJO,KO), TR(IOJO,KO), TC(IOJO,KO), TK(IKJK,K5), WELL(IOJO,KO
C          2), DELX(JO), DELY(IO), DELZ(KO), FACT(KO,3), PERM(IPJP), BOTTOM(
C          3IPJP), QRE(IQJQ), TF(3), A(IOJO), IN(6), XI(IOJO,KO), RM(2,MODE,
C          4ILJL,K7), JDIML1(IO), JDIML2(IO)        00002060
C
C      COMMON /INTEGR/ IO,JO,KO,I1,J1,K1,NPER,KTH,ITMAX,LENGTH,KP,NWEL,
C          1NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCH,
C          2IDK1,IDK2,IWATER,IQRE,IPJP,IQJQ,IKJK,K5,IPU1,IPU2,ITK,IEQN,MODE
C          .,ITL,IOJO,K6,K7,ILJL
C      COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,QR,DELM1        00002080

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

COMMON /SARRAY/ ICHK(13)                                00002090
COMMON /CK/ ETFLXT,STORT,QRET,CHST,CHDT,FLUXT,PUMPT,CFLUXT,FLXNT 00002100
COMMON /PR/ DIGIT(129),VF4(12),VF5(12),                  00002110
1VF6(12),VF7(12),VF8(12),VF9(12),VF10(12)
RETURN                                                 00002120
*****                                                 .
ENTRY DATAIN                                         00002130
*****                                                 .

---READ AND WRITE LIMIT OF ACTIVE NODES IN X DIRECTION

READ (5,330) (JDIML1(I),I=1,IO)                      00002140
READ (5,330) (JDIML2(I),I=1,IO)                      00002150
WRITE (6,460) (JDIML1(I),I=1,IO)                      00002160
WRITE (6,470) (JDIML2(I),I=1,IO)                      00002170

---READ AND WRITE SCALAR PARAMETERS---
READ (5,330) NPER,KTH,ERR,LENGTH                     00002180
WRITE (6,340) NPER,KTH,ERR                           00002190

---READ CUMULATIVE MASS BALANCE PARAMETERS---
READ (5,450) SUM,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETFL00002200
1XT,FLXNT,DELT
IF (IDK1.EQ.ICHK(4)) GO TO 20                         00002210
IF (IPU1.NE.ICHK(8)) GO TO 50                         00002220

---READ INITIAL HEAD VALUES FROM CARDS---
DO 15 K=1,K0                                           00002230
J4=1                                                 00002240
DO 10 I=1,IO                                           00002250
JJ=JO-JDIML2(I)                                       00002260
VF4(4)=DIGIT(129)                                     00002270
ND=(JDIML1(I)-1)*10                                    00002280
ND1=ND/80 + 123                                       00002290
ND=ND-80*(ND1-123)                                     00002300
J5=JDIML2(I)-JDIML1(I)+J4                            00002310
IF(ND1.EQ.123) ND1=128                               00002320
VF4(2)=DIGIT(ND1)                                     00002330
IF(ND.NE.0) GO TO 12                                 00002340
VF4(3)=DIGIT(128)                                     00002350
VF4(4)=DIGIT(128)                                     00002360
GO TO 11                                              00002370
12 VF4(3)=DIGIT(ND)                                   00002380
11 NE=(80-ND)/10                                      00002390
VF4(5)=DIGIT(NE)                                     00002400
READ (5,VF4) (PHI(IJ,K),IJ=J4,J5),(DELX(J),J=1,JJ) 00002410
10 J4=J5+1                                            00002420
15 CONTINUE                                           00002430
IF(ITL.NE.ICHK(12)) GO TO 16                         00002440
DO 17 K=1,K0                                           00002450

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

      READ(5,330) (XI(IJ,K),IJ=1,IOJO)          00002460
17  CONTINUE                                     00002470
      READ(5,480) (((RM(N,M,IJ,K),N=1,2),M=1,MODE),IJ=1,ILJL),K=1,K7) 00002480
16  CONTINUE                                     00002490
      GO TO 30                                     00002500

C
C   ---READ INITIAL HEAD AND MASS BALANCE PARAMETERS FROM DISK---
20  READ (4) PHI,SUM,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETFL00002510
    1XT,FLXNT,DELT,XI,RM
      REWIND 4                                     00002520
30  WRITE (6,430) SUM                           00002530
    DO 45 K=1,K0                                 00002540
    J4=1                                         00002550
    WRITE (6,440) K                           00002560
    DO 40 I=1,IO                               00002570
    ND=(JDIML1(I)-1)*6                         00002580
    ND1=ND/120 + 123                          00002590
    VF5(2)=DIGIT(ND1)                         00002600
    ND=ND-120*(ND1-123)                      00002610
    NE=(120-ND)/6                            00002620
    J5=JDIML2(I)-JDIML1(I)+J4                00002630
    IF(ND1.NE.123) ND=3+ND                     00002640
    ND=ND+2                                     00002650
    VF5(3)=DIGIT(ND)                         00002660
    VF5(5)=DIGIT(NE)                         00002670
    WRITE (6,VF5) I,(PHI(IJ,K),IJ=J4,J5)     00002680
40  J4=J5+1                                    00002690
45  CONTINUE                                     00002700

C
50  DO 60 K=1,K0                                00002710
    DO 60 IJ=1,IOJO                            00002720
    WELL(IJ,K)=0.                             00002730
    TR(IJ,K)=0.                               00002740
    TC(IJ,K)=0.                               00002750
    IF (K.NE.K0) TK(IJ,K)=0.                  00002760
60  CONTINUE                                     00002770
    RETURN                                       00002780
C
C   ****
C   ENTRY ARRAY(A,INFT,IOFT,IN,IRN,TF,K)        00002790
C
C   ****
    READ (5,330) FAC,IVAR,IPRN,TF,IRECS,IRECD 00002800
    IC=4*IRECS+2*IVAR+IPRN+1                   00002810
    GO TO (70,70,90,90,120,120), IC           00002820
70  DO 80 IJ=1,IOJO                            00002830
80  A(IJ)=FAC                                  00002840
    WRITE (6,280) IN,FAC,K                     00002850
    GO TO 140                                   00002860
90  IF (IC.EQ.3) WRITE (6,290) IN,K           00002870
    J4=1                                         00002880
    DO 111 I=1,IO                           00002890
    JJ=JO-JDIML2(I)                         00002900

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

IF (INFT.EQ.1) GO TO 91          00002910
VF4(4)=DIGIT(129)                00002920
ND=(JDIML1(I)-1)*10              00002930
ND1=ND/80 + 123                  00002940
ND=ND-80*(ND1-123)                00002950
J5=JDIML2(I)-JDIML1(I)+J4        00002960
IF(ND1.EQ.123) ND1=128            00002970
VF4(2)=DIGIT(ND1)                00002980
IF(ND.NE.0) GO TO 71              00002990
VF4(3)=DIGIT(128)                00003000
VF4(4)=DIGIT(128)                00003010
GO TO 72                          00003020
71 VF4(3)=DIGIT(ND)                00003030
72 NE=(80-ND)/10                  00003040
VF4(5)=DIGIT(NE)                  00003050
DO 89 KVF=1,12                   00003060
89 INFT1(KVF)=VF4(KVF)            00003070
GO TO 92                          00003080
91 ND=(JDIML1(I)-1)*4              00003090
VF6(4)=DIGIT(129)                00003100
ND1=ND/80 + 123                  00003110
ND=ND-80*(ND1-123)                00003120
J5=JDIML2(I)-JDIML1(I)+J4        00003130
IF(ND1.EQ.123) ND1=128            00003140
VF6(2)=DIGIT(ND1)                00003150
IF(ND.NE.0) GO TO 73              00003160
VF6(3)=DIGIT(128)                00003170
VF6(4)=DIGIT(128)                00003180
GO TO 74                          00003190
73 VF6(3)=DIGIT(ND)                00003200
74 NE=(80-ND)/4                  00003210
IF(JO.LE.20) NE=NE+JJ              00003220
VF6(5)=DIGIT(NE)                  00003230
DO 88 KVF=1,12                   00003240
88 INFT1(KVF)=VF6(KVF)            00003250
92 READ (5,INFT1) (A(IJ),IJ=J4,J5),(DELX(J),J=1,JJ) 00003260
DO 100 IJ=J4,J5                  00003270
100 A(IJ)=A(IJ)*FAC               00003280
GO TO (101,102,103,104) ,IOFT      00003290
101 ND=(JDIML1(I)-1)*6             00003300
ND1=ND/120 + 123                  00003310
VF5(2)=DIGIT(ND1)                00003320
ND=ND-120*(ND1-123)                00003330
NE=(120-ND)/6                     00003340
IF(ND1.NE.123) ND=3+ND              00003350
ND=ND+2                            00003360
VF5(3)=DIGIT(ND)                  00003370
VF5(5)=DIGIT(NE)                  00003380
DO 87 KVF=1,12                   00003390
87 IOFT1(KVF)=VF5(KVF)            00003400
GO TO 110                          00003410

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

102 ND=(JDIML1(I)-1)*9          00003420
    ND1=ND/126 + 123            00003430
    VF7(2)=DIGIT(ND1)          00003440
    ND=ND - 126*(ND1 - 123)    00003450
    NE=(126-ND)/9              00003460
    IF(ND1.NE.123) ND=6+ND      00003470
    VF7(3)=DIGIT(ND)           00003480
    VF7(5)=DIGIT(NE)           00003490
        DO 86 KVF=1,12          00003500
86  IOFT1(KVF)=VF7(KVF)        00003510
    GO TO 110                  00003520
103 ND=(JDIML1(I)-1)*12        00003530
    ND1=ND/120 + 123            00003540
    VF8(2)=DIGIT(ND1)          00003550
    ND=ND - 120*(ND1 - 123)    00003560
    NE=(120 - ND)/12           00003570
    IF(ND1.NE.123) ND=6+ND      00003580
    VF8(3)=DIGIT(ND)           00003590
    VF8(5)=DIGIT(NE)           00003600
        DO 85 KVF=1,12          00003610
85  IOFT1(KVF)=VF8(KVF)        00003620
    GO TO 110                  00003630
104 ND=(JDIML1(I)-1)*11        00003640
    ND1=ND/110 + 123            00003650
    VF9(2)=DIGIT(ND1)          00003660
    ND=ND-110*(ND1-123)        00003670
    NE=(110-ND)/11              00003680
    IF(ND1.NE.123) ND=6+ND      00003690
    VF9(3)=DIGIT(ND)           00003700
    VF9(5)=DIGIT(NE)           00003710
        DO 105 KVF=1,12          00003720
105 IOFT1(KVF)=VF9(KVF)        00003730
110 IF (IC.EQ.3) WRITE (6,IOFT1) I,(A(IJ),IJ=J4,J5) 00003740
111 J4=J5+1                      00003750
    GO TO 140                  00003760
120 READ (2'IRN) A              00003770
    IF (IC.EQ.6) GO TO 140      00003780
    WRITE (6,290) IN,K          00003790
    J4=1                         00003800
    DO 131 I=1,I0                00003810
    GO TO (121,122,123,124) ,IOFT 00003820
121 ND=(JDIML1(I)-1)*6          00003830
    ND1=ND/120 + 123            00003840
    VF5(2)=DIGIT(ND1)          00003850
    ND=ND-120*(ND1-123)        00003860
    NE=(120-ND)/6              00003870
    IF(ND1.NE.123) ND=3+ND      00003880
    ND=ND+2                      00003890
    VF5(3)=DIGIT(ND)           00003900
    VF5(5)=DIGIT(NE)           00003910
        DO 119 KVF=1,12          00003920

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

119 IOFT1(KVF)=VF5(KVF)          00003930
    GO TO 130                    00003940
122 ND=(JDIML1(I)-1)*9          00003950
    ND1=ND/126 + 123             00003960
    VF7(2)=DIGIT(ND1)            00003970
    ND=ND-126*(ND1-123)          00003980
    NE=(126-ND)/9               00003990
    IF(ND1.NE.123) ND=6+ND       00004000
    VF7(3)=DIGIT(ND)             00004010
    VF7(5)=DIGIT(NE)             00004020
        DO 118 KVF=1,12           00004030
118 IOFT1(KVF)=VF7(KVF)          00004040
    GO TO 130                    00004050
123 ND=(JDIML1(I)-1)*12          00004060
    ND1=ND/120 + 123             00004070
    VF8(2)=DIGIT(ND1)            00004080
    ND=ND-120*(ND1-123)          00004090
    NE=(120-ND)/12               00004100
    IF(ND1.NE.123) ND=6+ND       00004110
    VF8(3)=DIGIT(ND)             00004120
    VF8(5)=DIGIT(NE)             00004130
        DO 117 KVF=1,12           00004140
117 IOFT1(KVF)=VF8(KVF)          00004150
    GO TO 130                    00004160
124 ND=(JDIML1(I)-1)*11          00004170
    ND1=ND/110 + 123             00004180
    VF9(2)=DIGIT(ND1)            00004190
    ND=ND-110*(ND1-123)          00004200
    NE=(110-ND)/11               00004210
    IF(ND1.NE.123) ND=6+ND       00004220
    VF9(3)=DIGIT(ND)             00004230
    VF9(5)=DIGIT(NE)             00004240
        DO 125 KVF=1,12           00004250
125 IOFT1(KVF)=VF9(KVF)          00004260
130 WRITE (6,IOFT1) I,(A(IJ),IJ=J4,J5) 00004270
131 J4=J5+1                      00004280
140 IF (IRECD.EQ.1) WRITE (2'IRN) A   00004290
    IRN=IRN+1                    00004300
    RETURN                         00004310
C ****
C ENTRY MDAT                     00004320
C ****
DO 155 K=1,K0                  00004330
J4=1                           00004340
DO 152 I=1,I0                  00004350
J5=JDIML2(I)-JDIML1(I)+J4      00004360
DO 150 IJ=J4,J5                00004370
IF (I.EQ.1.OR.I.EQ.I0) T(IJ,K)=0. 00004380
IF (IDK1.NE.ICHK(4).AND.IPU1.NE.ICHK(8)) PHI(IJ,K)=STRT(IJ,K) 00004390
IF (K.NE.K0.OR.IWATER.NE.ICHK(6)) GO TO 150 00004400
IF (I.EQ.1.OR.I.EQ.I0) PERM(IJ)=0. 00004410

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

150  CONTINUE          00004420
152  J4=J5+1          00004430
155  CONTINUE          00004440
C   ..... DELX ..... 00004450
      DO 153 J=1,J0    00004460
153  DELX(J)=0.       00004470
      READ (5,330) FAC,IVAR,IPRN 00004480
      IF (IVAR.EQ.1) READ (5,330) (DELX(J),J=1,J0)
      DO 170 J=1,J0    00004490
      IF (IVAR.NE.1) GO TO 160 00004500
      DELX(J)=DELX(J)*FAC 00004510
      GO TO 170          00004520
160  DELX(J)=FAC     00004530
170  CONTINUE          00004540
      IF (IVAR.EQ.1.AND.IPRN.NE.1) WRITE (6,370) (DELX(J),J=1,J0) 00004550
      IF (IVAR.EQ.0) WRITE (6,300) FAC 00004560
C   ..... DELY ..... 00004570
      READ (5,330) FAC,IVAR,IPRN 00004580
      IF (IVAR.EQ.1) READ (5,330) (DELY(I),I=1,IO)
      DO 190 I=1,IO    00004590
      IF (IVAR.NE.1) GO TO 180 00004600
      DELY(I)=DELY(I)*FAC 00004610
      GO TO 190          00004620
180  DELY(I)=FAC     00004630
190  CONTINUE          00004640
      IF (IVAR.EQ.1.AND.IPRN.NE.1) WRITE (6,380) (DELY(I),I=1,IO) 00004650
      IF (IVAR.EQ.0) WRITE (6,310) FAC 00004660
C   ..... DELZ ..... 00004670
      READ (5,330) FAC,IVAR,IPRN 00004680
      IF (IVAR.EQ.1) READ (5,330) (DELZ(K),K=1,K0)
      DO 210 K=1,K0    00004690
      IF (IVAR.NE.1) GO TO 200 00004700
      DELZ(K)=DELZ(K)*FAC 00004710
      GO TO 210          00004720
200  DELZ(K)=FAC     00004730
210  CONTINUE          00004740
      IF (IVAR.EQ.1.AND.IPRN.NE.1) WRITE (6,390) (DELZ(K),K=1,K0) 00004750
      IF (IVAR.EQ.0) WRITE (6,320) FAC 00004760
C   ---INITIALIZE VARIABLES--- 00004770
      B=0.                00004780
      D=0.                00004790
      F=0.                00004800
      H=0.                00004810
      SU=0.               00004820
      Z=0.                00004830
      RETURN
C   ..... 00004830
C   ---READ TIME PARAMETERS AND PUMPING DATA FOR A NEW PUMPING PERIOD-

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

*****  

ENTRY NEWPER                                00004840
*****  

DELTM1 = DELT                                00004850
READ (5,330) KP,KPM1,NWEL,TMAX,NUMT,CDLT,DELT 00004860
IF (KP.GT.1) GO TO 225                      00004870
IF(ITL.EQ.ICHK(12).AND.(IDK1.EQ.ICHK(4).OR.IPU1.EQ.ICHK(8))) GO TO00004880
1 225
DELTM1 = 1.0                                  00004890
DO 215 K=1,K0                                 00004900
DO 215 IJ=1,IOJO                            00004910
215 XI(IJ,K) = 0.0                          00004920
C
C   ---COMPUTE ACTUAL DELT AND NUMT---
225 DT=DELT/24.                               00004930
TM=0.0                                         00004940
DO 220 I=1,NUMT                             00004950
DT=CDLT*DT                                    00004960
TM=TM+DT                                     00004970
IF (TM.GE.TMAX) GO TO 230                  00004980
220 CONTINUE                                 00004990
GO TO 240                                   00005000
230 DELT=TMAX/TM*DELT                      00005010
NUMT=I                                       00005020
240 WRITE (6,400) KP,TMAX,NUMT,DELT,CDLT    00005030
DELT=DELT*3600.                            00005040
TMAX=TMAX*86400.                           00005050
SUMP=0.0                                     00005060
C
C   ---READ AND WRITE WELL PUMPING RATES---
WRITE (6,410) NWEL                           00005070
IF (NWEL.EQ.0) GO TO 260                     00005080
DO 245 K=1,K0                                 00005090
DO 245 IJ=1,IOJO                            00005100
245 WELL(IJ,K)=0.0                          00005110
DO 250 II=1,NWEL                            00005120
READ (5,330) K,I,J,TWELL                   00005130
IJ=0                                         00005140
J4=1                                         00005150
IV=I-1                                      00005160
DO 246 III=1,IV                            00005170
J5=JDIML2(III)-JDIML1(III)+J4              00005180
246 J4=J5+1                                 00005190
IJ=J4+J-JDIML1(I)                         00005200
WRITE (6,420) K,I,J,TWELL                   00005210
250 WELL(IJ,K)=TWELL/(DELX(J)*DELY(I))    00005220
260 RETURN                                  00005230
C
C   ---FORMATS---

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

280 FORMAT (1H0,52X,6A4,' = ',1P1G15.7,' FOR LAYER',I3)          00005240
290 FORMAT (1H1,45X,6A4,' MATRIX, LAYER',I3/46X,41('-'))          00005250
300 FORMAT ('0',72X,'DELX = ',1P1G15.7)                            00005260
310 FORMAT ('0',72X,'DELY = ',1P1G15.7)                            00005270
320 FORMAT ('0',72X,'DELZ = ',1P1G15.7)                            00005280
330 FORMAT (8G10.0)                                                 00005290
340 FORMAT ('0',51X,'NUMBER OF PUMPING PERIODS = ',I5/49X,'TIME STEPS B00005300
1ETWEEN PRINTOUTS = ',I5//51X,'ERROR CRITERIA FOR CLOSURE = ',1P1G15.
.7)
370 FORMAT (1H1,46X,40HGRID SPACING IN PROTOTYPE IN X DIRECTION/47X,4000005310
1('-')//(0',12F10.0))
380 FORMAT (1H-,46X,40HGRID SPACING IN PROTOTYPE IN Y DIRECTION/47X,4000005320
1('-')//(0',12F10.0))
390 FORMAT (1H-,46X,40HGRID SPACING IN PROTOTYPE IN Z DIRECTION/47X,4000005330
1('-')//(0',12F10.0))
400 FORMAT ('-',50X,'PUMPING PERIOD NO.',I4,':',F10.2,' DAYS'/51X,38('00005340
1-')//53X,'NUMBER OF TIME STEPS=',I6//59X,'DELT IN HOURS =',F10.3//'
253X,'MULTIPLIER FOR DELT =',F10.3)
410 FORMAT ('-',63X,I4,' WELLS'/65X,9('-')//50X,'K',9X,'I',9X,'J      PU00005350
1MPING RATE')
420 FORMAT (41X,3I10,2F13.2)                                         00005360
430 FORMAT ('-',40X,' CONTINUATION - HEAD AFTER ',G20.7,' SEC PUMPING 00005370
1'/42X,58('-'))
440 FORMAT ('1',55X,'INITIAL HEAD MATRIX, LAYER',I3/56X,30('-'))    00005380
450 FORMAT (4G20.10)                                                 00005390
460 FORMAT(1H1,46X,'POSITION OF FIRST ACTIVE NODE IN THE X DIRECTION'/00005400
147X,44('-')//(0',12I10))
470 FORMAT(1H0,46X,'POSITION OF LAST ACTIVE NODE IN THE X DIRECTION'/ 00005410
147X,43('-')//(0',12I10))
480 FORMAT(6E13.6)                                                 00005420
END                                                               00005430
SUBROUTINE STEP(PHI,STRT,OLD,T,S,TR,TC,TK,WELL,DELX,DELY,DELZ,FACT00005440
1,DDN,TEST3,XI,RM,JDIML1,JDIML2)
-----
INITIALIZE DATA FOR A NEW TIME STEP AND PRINT RESULTS
-----
SPECIFICATIONS:
REAL *8PHI                                                       00005450
DIMENSION PHI(IOJO,K0), STRT(IOJO,K0), OLD(IOJO,K0), T(IOJO,K0      00005460
1), S(IOJO,K0), TR(IOJO,K0), TC(IOJO,K0), TK(IKJK,K5), WELL(IOJO
2,K0), DELX(J0), DELY(IO), DELZ(K0), FACT(KO,3), DDN(IMAX), TEST3
3(ITMX1), ITTO(50)
4,XI(IOJO,K0),RM(2,MODE,ILJL,K7),JDIML1(IO),JDIML2(IO)
COMMON /INTEGR/ IO,JO,K0,I1,J1,K1,NPER,KTH,ITMAX,LENGTH,KP,NWEL,    00005470
1NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCH,
2IDK1,IDK2,IWATER,IQRE,IPJP,IQJQ,IKJK,K5,IPU1,IPU2,ITK,IEQN,MODE

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

. , ITL, IOJO, K6, K7, ILJL
COMMON /SPARAM/ TMAX, CDLT, DELT, ERR, TEST, SUM, SUMP, QR, DELTM1      00005480
COMMON /SARRAY/ ICHK(13)                                                 00005490
COMMON /CK/ ETFLXT, STORT, QRET, CHST, CHDT, FLUXT, PUMPT, CFLUXT, FLXNT 00005500
COMMON /PR/ DIGIT(129), VF4(12), VF5(12),
1VF6(12), VF7(12), VF8(12), VF9(12), VF10(12)                           00005510
      RETURN                                                               00005520
C
C ***** *****
C ENTRY NEWSTP                                                       00005530
C ***** *****
C KT=KT+1                                                               00005540
C IT=0                                                               00005550
C IF (KT.GT.1)    DELTM1 = DELT                                         00005560
C DELT=CDLT*DELT                                         00005570
C SUM=SUM+DELT                                         00005580
C SUMP=SUMP+DELT                                         00005590
C DAYSP=SUMP/86400.                                         00005600
C YRSP=DAYSP/365.                                         00005610
C HRS=SUM/3600.                                         00005620
C SMIN=HRS*60.                                         00005630
C DAYS=HRS/24.                                         00005640
C YRS=DAYS/365.                                         00005650
C .....COMPUTE TRANSIENT LEAKAGE COEFFICIENTS.....
C IF (ITL.NE.ICHK(12)) GO TO 11                                         00005660
C CALL CLAY                                                               00005670
C 11 DO 10 K=1,K0                                         00005680
C     DO 10 IJ=1,IOJO                                         00005690
C     XI(IJ,K) = 0.0                                         00005700
C 10 OLD(IJ,K)=PHI(IJ,K)                                         00005710
C     RETURN                                                               00005720
C
C ---PRINT OUTPUT AT DESIGNATED TIME STEPS---
C ***** *****
C ENTRY OUTPUT                                                       00005730
C ***** *****
C IF (KT.EQ.NUMT) IFINAL=1                                         00005740
C ITTO(KT)=IT                                         00005750
C IF (IT.LE.ITMAX) GO TO 20                                         00005760
C IT=IT-1                                         00005770
C ITTO(KT)=IT                                         00005780
C IERR=2                                                               00005790
C
C ---IF MAXIMUM ITERATIONS EXCEEDED, WRITE RESULTS ON DISK OR CARDS--
C IF (IDK2.EQ.ICHK(5)) WRITE (4) PHI, SUM, SUMP, PUMPT, CFLUXT, QRET, CHST 00005800
C 1, CHDT, FLUXT, STORT, ETFLXT, FLXNT, DELT, XI, RM
C   IF (IPU2.EQ.ICHK(9)) WRITE (7,230) SUM, SUMP, PUMPT, CFLUXT, QRET, CHST 00005810
C 1, CHDT, FLUXT, STORT, ETFLXT, FLXNT, DELT
C
C 20 IF (IFLO.EQ.ICHK(3)) CALL CHECK                                00005820
C     IF (IERR.EQ.2) GO TO 30                                         00005830

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

IF (MOD(KT,KTH).NE.0.AND.IFINAL.NE.1) RETURN          00005840
30 WRITE (6,210) KT,DELT,SUM,SMIN,HRS,DAYS,YRS,DAYSP,YRSP 00005850
IF (IFLO.EQ.ICHK(3)) CALL CWRITE                      00005860
IT=IT+1                                              00005870
WRITE (6,180) (TEST3(J),J=1,IT)                      00005880
I3=1                                                00005890
I5=0                                                00005900
352 I5=I5+40                                         00005910
I4=MIN0(KT,I5)                                      00005920
WRITE (6,240) (I,I=I3,I4)                           00005930
WRITE (6,260)                                         00005940
WRITE (6,250) (ITTO(I),I=I3,I4)                      00005950
WRITE (6,260)                                         00005960
IF(KT.LE.I5) GO TO 353                            00005970
I3=I3+40                                           00005980
GO TO 352                                         00005990
C
353 IF (IDRAW.NE.ICHK(1)) GO TO 100                00006000
C
C ---PRINT DRAWDOWN---
DO 90 K=1,K0                                         00006010
J4=1                                              00006020
WRITE (6,200) K                                       00006030
DO 90 I=1,IO                                         00006040
J=0                                               00006050
J5=JDIML2(I)-JDIML1(I)+J4                          00006060
ND=(JDIML1(I)-1)*7                                 00006070
ND1=ND/126 + 123                                    00006080
VF10(2)=DIGIT(ND1)                                00006090
ND=ND-126*(ND1-123)                               00006100
NE=(126-ND)/7                                     00006110
IF(ND1.NE.123) ND=5+ND                            00006120
VF10(3)=DIGIT(ND)                                00006130
VF10(5)=DIGIT(NE)                                00006140
DO 80 IJ=J4,J5                                     00006150
J=J+1                                              00006160
80 DDN(J)=STRT(IJ,K)-PHI(IJ,K)                  00006170
J4=J5+1                                           00006180
90 WRITE (6,VF10) I,(DDN(II),II=1,J)             00006190
100 IF (IHEAD.NE.ICHK(2)) GO TO 120            00006200
C
C ---PRINT HEAD MATRIX---
DO 115 K=1,K0                                         00006210
J4=1                                              00006220
WRITE (6,190) K                                       00006230
DO 110 I=1,IO                                         00006240
J5=JDIML2(I)-JDIML1(I)+J4                          00006250
ND=(JDIML1(I)-1)*7                                 00006260
ND1=ND/126 + 123                                    00006270
VF10(2)=DIGIT(ND1)                                00006280
ND=ND-126*(ND1-123)                               00006290

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

NE=(126-ND)/7                                00006300
IF(ND1.NE.123) ND=5+ND                      00006310
VF10(3)=DIGIT(ND)                           00006320
VF10(5)=DIGIT(NE)                           00006330
WRITE (6, VF10) I,(PHI(IJ,K),IJ=J4,J5)     00006340
110 J4=J5+1                                 00006350
115 CONTINUE                               00006360
C
C   ---WRITE ON DISK---
120 IF (IERR.EQ.2) GO TO 130                 00006370
    IF (KP.LT.NPER.OR.IFINAL.NE.1) RETURN      00006380
    IF (IDK2.EQ.1CHK(5)) WRITE (4) PHI,SUM,SUMP,PUMPT,CFLUXT,QRET,CHST00006390
    1,CHDT,FLUXT,STORT,ETFLXT,FLXNT,DELT,XI,RM
C
C   ---PUNCHED OUTPUT---
130 IF (IPU2.NE.1CHK(9)) GO TO 160          00006400
    IF (IERR.EQ.2) GO TO 140                  00006410
    WRITE (7,230) SUM,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETF00006420
    1LXT,FLXNT,DELT
140 DO 135 J=1,JO                            00006430
135 DELX(J)=0.0                             00006440
    DO 155 K=1,K0                           00006450
    J4=1                                  00006460
    DO 150 I=1,I0                           00006470
    JJ=JO-JDIML2(I)                         00006480
    VF4(4)=DIGIT(129)                       00006490
    J5=JDIML2(I)-JDIML1(I)+J4              00006500
    ND=(JDIML1(I)-1)*10                     00006510
    ND1=ND/80 + 123                         00006520
    ND=ND-80*(ND1-123)                      00006530
    IF(ND1.EQ.123) ND1=128                  00006540
    VF4(2)=DIGIT(ND1)                       00006550
    IF(ND.NE.0) GO TO 141                  00006560
    VF4(4)=DIGIT(128)                       00006570
    VF4(3)=DIGIT(128)                       00006580
    GO TO 142                                00006590
141 VF4(3)=DIGIT(ND)                         00006600
142 NE=(80-ND)/10                           00006610
    VF4(5)=DIGIT(NE)                         00006620
    WRITE (7,VF4) (PHI(IJ,K),IJ=J4,J5),(DELX(J),J=1,JJ) 00006630
150 J4=J5+1                                 00006640
155 CONTINUE                               00006650
    IF(ITL.NE.1CHK(12)) GO TO 160          00006660
    DO 156 K=1,K0                           00006670
    WRITE(7,270) (XI(IJ,K),IJ=1,IOJO)       00006680
156 CONTINUE                               00006690
    WRITE(7,280) (((RM(N,M,IJ,K),N=1,2),M=1,MODE),IJ=1,ILJL),K=1,K7) 00006700
160 IF (IERR.EQ.2) STOP                      00006710
    RETURN                                 00006720
C
C   ---FORMATS---

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

C  
C  
C

```

180 FORMAT ('OMAXIMUM HEAD CHANGE FOR EACH ITERATION:'//',39('-')/('000006730
1',10F12.4))
190 FORMAT ('1',55X,'HEAD MATRIX, LAYER',I3/56X,21('-')) 00006740
200 FORMAT ('1',55X,' DRAWDOWN, LAYER',I3/59X,18('-')) 00006750
210 FORMAT (1H1,44X,57('-')/45X,'±',14X,'TIME STEP NUMBER =',I9,14X,'±00006760
1'/45X,57('-')//50X,29HSIZE OF TIME STEP IN SECONDS=,F14.2//55X,'TO
2TAL SIMULATION TIME IN SECONDS=',F14.2/80X,8HMINUTES=,F14.2/82X,6H
3HOURS=,F14.2/83X,5HDAYS=,F14.2/82X,'YEARS=',F14.2///45X,'DURATION
4OF CURRENT PUMPING PERIOD IN DAYS=',F14.2/82X,'YEARS=',F14.2//) 00006770
230 FORMAT (1P4G20.10) 00006780
240 FORMAT ('OTIME STEP :',40I3) 00006790
250 FORMAT ('OITERATIONS:',40I3) 00006800
260 FORMAT (' ',10('-')) 00006810
270 FORMAT(8F10.6) 00006820
280 FORMAT(6E13.6)
END 00006830
SUBROUTINE SOLVE(PHI,STRT,OLD,T,S,TR,TC,TK,WELL,DELX,DELY,DELZ,FAC00006840
1T,EL,FL,GL,V,XI,TEST3,QRE,RATE,ZCB,SS,TL,TLK,SL,RM,JDIML1,JDIML2)
-----
```

C  
C  
C  
C

SPECIFICATIONS:

```

REAL *8PHI,RHO,B,D,F,H,Z,SU,RHOP,W,WMIN,RHO1,RHO2,RHO3,XPART,YPART00005
1,ZPART,DMIN1,WMAX,XT,YT,ZT,DABS,DMAX1,DEN,TXM,TYM,TZM
REAL *8E,AL,BL,CL,A,C,G,WU,TU,U,DL,RES,SUPH,GLXI,ZPHI 00006840
C
```

```

DIMENSION PHI(1), STRT(1), OLD(1), T(1), S(1), TR(1), TC(1), TK(1)00006870
1, WELL(1), DELX(1), DELY(1), DELZ(1), FACT(K0,3), RHOP(20), TEST3(
21), EL(1), FL(1), GL(1), V(1), XI(1), QRE(1), RATE(1), ZCB(1),
3SS(1), TL(1), TLK(1), SL(1), RM(1), JDIML1(1), JDIML2(1)
```

C

```

COMMON /INTEGR/ IO,JO,K0,I1,J1,K1,NPER,KTH,ITMAX,LENGTH,KP,NWEL, 00006880
1NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCH,
2IDK1,IDK2,IWATER,IQRE,IPJP,IQJQ,IKJK,K5,IPU1,IPU2,ITK,IEQN,MODE
.,ITL,IOJO,K6,K7,ILJL
COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,QR,DELM1 00006890
COMMON /SARRAY/ ICHK(13) 00006900
RETURN 00006910
C
```

C  
C

\*\*\*\*\*

ENTRY ITER 00006920

\*\*\*\*\*

---COMPUTE AND PRINT ITERATION PARAMETERS---

WRITE (6,240) 00006930

P2=LENGTH-1 00006940

NT=IOJO\*K0 00006950

NIJ=IOJO 00006960

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

READ 888,WMAX          00006970
888 FORMAT(F10.0)      00006980
PRINT 889,WMAX         00006990
889 FORMAT(5X,6HWMAX =,F10.7) 00007000
PJ=-1.                 00007010
DO 50 I=1,LENGTH       00007020
PJ=PJ+1.               00007030
50 RHOP(I)=1.D0-(1.D0-WMAX)**(PJ/P2) 00007040
WRITE (6,230) LENGTH,(RHOP(J),J=1,LENGTH) 00007050
RETURN                00007060
C
C
C ---INITIALIZE DATA FOR A NEW ITERATION---
60 IT=IT+1              00007070
IF (IT.LE.ITMAX) GO TO 70 00007080
WRITE (6,220)             00007090
CALL OUTPUT              00007100
70 IF (MOD(IT,LENGTH)) 80,80,90 00007110
C
C ENTRY NEWITA           00007120
C
C
80 NTH=0                00007130
90 NTH=NTH+1             00007140
W=RHOP(NTH)              00007150
TEST3(IT+1)=0.            00007160
TEST=0.0                 00007170
BIG=0.                   00007180
DO 100 I=1,NT             00007190
EL(I)=0.                 00007200
FL(I)=0.                 00007210
GL(I)=0.                 00007220
100 V(I)=0.               00007230
C
C ---COMPUTE TRANSMISSIVITY AND T COEFFICIENTS FOR UPPER
C HYDROLOGIC UNIT WHEN IT IS UNCONFINED---
IF (IWATER.NE.ICHK(6)) GO TO 110 00007240
CALL TRANS(0)             00007250
C
C ---CHOOSE SIP NORMAL OR REVERSE ALGORITHM---
110 IF (MOD(IT,2)) 120,120,170 00007260
120 J9=1                 00007270
DO 155 K=1,K0             00007280
J4=JDIML2(1)-JDIML1(1)+2 00007290
KN=(K-1)*NIJ              00007300
DO 152 I=2,I1             00007310
J5=JDIML2(I)-JDIML1(I)+J4 00007320
J6=J4+KN                 00007330
J7=J5+KN                 00007340
J8=J7+JDIML2(I+1)-JDIML1(I+1)+1 00007350
DO 150 IJ=J4,J5             00007360
N=IJ+KN                  00007370

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

C
C   ---SKIP COMPUTATIONS IF NODE CONSTANT HEAD---
C
C   IF(S(N).LT.0.) GO TO 150                                00007380
C
NIB=N-JDIML2(I-1)+JDIML1(I)-1                                00007390
NIA=N+JDIML2(I)-JDIML1(I+1)+1                                00007400
NJA=N+1                                                       00007410
NJB=N-1                                                       00007420
  IF(N.EQ.J6) NJB=1                                         00007430
  IF(N.EQ.J7) NJA=1                                         00007440
  IF(NIB.GE.J6)NIB=1                                         00007450
  IF(NIB.LT.J9) NIB=1                                         00007460
  IF(NIA.GT.J8)NIA=1                                         00007470
  IF(NIA.LE.J7) NIA=1                                         00007480
  IF(I.EQ.2) NIB=1                                           00007490
  IF(I.EQ.I1) NIA=1                                         00007500
NKA=N+NIJ                                                       00007510
NKB=N-NIJ                                                       00007520
C
C
IF (ITL.NE.ICHK(12)) GO TO 121                                00007530
L=N                                                       00007540
LKB=NKB                                                       00007550
GO TO 122                                                       00007560
121   L=1                                                       00007570
LKB=1                                                       00007580
TLK(LKB)=0.                                                 00007590
TLK(L)=0.                                                 00007600
TL(L)=0.                                                 00007610
SL(L)=0.                                                 00007620
C
C   ---SKIP COMPUTATIONS IF NODE OUTSIDE FLOW SYSTEM---
C
122   IF(K.EQ.1) GO TO 111                                00007630
  IF(K.EQ.K0) GO TO 113                                00007640
  IF(T(N).EQ.0..AND.TK(N).EQ.0..AND.TK(NKB).EQ.0..AND.TLK(L).EQ.0.
1 .AND.TLK(LKB).EQ.0.) GO TO 150                         00007650
  GO TO 112                                               00007660
111   IF(T(N).EQ.0..AND.TK(N).EQ.0..AND.TLK(L).EQ.0.) GO TO 150 00007670
  GO TO 112                                               00007680
113   IF(T(N).EQ.0..AND.TK(NKB).EQ.0..AND.TLK(LKB).EQ.0.) GO TO 150 00007690
C   ---COMPUTE COEFFICIENTS---
112   J=N-J4-KN+JDIML1(I)                                 00007700
  D=TR(NJB)/DELX(J)                                     00007710
  F=TR(N)/DELX(J)                                      00007720
  B=TC(NIB)/DELY(I)                                     00007730
  H=TC(N)/DELY(I)                                      00007740
  SU=0.DO                                              00007750
  Z=0.DO                                              00007760
  IF(K.EQ.1) GO TO 124                                00007770

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

Z=TK(NKB) + TLK(LKB) 00007780
IF(IEQN.EQ.ICHK(11)) Z=Z/DELZ(K) 00007790
124 IF(K.EQ.KO) GO TO 125 00007800
SU=TK(N) + TLK(L) 00007810
IF(IEQN.EQ.ICHK(11))SU=SU/DELZ(K) 00007820
125 RHO=S(N)/DELT 00007830
QR=0. 00007840
IF (K.EQ.KO.AND.IQRE.EQ.ICHK(7)) QR=QRE(IJ) 00007850

---SIP NORMAL ALGORITHM---
---FORWARD SUBSTITUTE, COMPUTING INTERMEDIATE VECTOR V---
IF (K.EQ.1) GO TO 131 00007860
IF (K.EQ.KO) GO TO 133 00007870
GO TO 130 00007880
131 E=-B-D-F-H-SU-Z-RHO-TL(L)+TLK(L) 00007890
GO TO 132 00007900
133 E=-B-D-F-H-SU-Z-RHO+TLK(LKB)-TL(L) 00007910
GO TO 132 00007920
130 E=-B-D-F-H-SU-Z-RHO+TLK(LKB)+TLK(L)-TL(L) 00007930
132 CONTINUE 00007940
BL=B/(1.+W*(EL(NIB)+GL(NIB))) 00007950
CL=D/(1.+W*(FL(NJB)+GL(NJB))) 00007960
C=BL*EL(NIB) 00007970
G=CL*FL(NJB) 00007980
WU=CL*GL(NJB) 00007990
U=BL*GL(NIB) 00008000
IF (K.EQ.1) GO TO 140 00008010
AL=Z/(1.+W*(EL(NKB)+FL(NKB))) 00008020
A=AL*EL(NKB) 00008030
TU=AL*FL(NKB) 00008040
DL=E+W*(A+C+G+WU+TU+U)-CL*EL(NJB)-BL*FL(NIB)-AL*GL(NKB) 00008050
EL(N)=(F-W*(A+C))/DL 00008060
FL(N)=(H-W*(G+TU))/DL 00008070
GL(N)=(SU-W*(WU+U))/DL 00008080
SUPH=0.DO 00008090
IF (K.NE.KO) SUPH=SU*PHI(NKA) 00008100
RES=-B*PHI(NIB)-D*PHI(NJB)-E*PHI(N)-F*PHI(NJA)-H*PHI(NIA)-SUPH-Z*P00008110
1HI(NKB)-WELL(N)-RHO*OLD(N)-QR-SL(L)
V(N)=(RES-AL*V(NKB)-BL*V(NIB)-CL*V(NJB))/DL 00008120
GO TO 150 00008130
140 DL=E+W*(C+G+WU+U)-CL*EL(NJB)-BL*FL(NIB) 00008140
EL(N)=(F-W*C)/DL 00008150
FL(N)=(H-W*G)/DL 00008160
GL(N)=(SU-W*(WU+U))/DL 00008170
SUPH=0.DO 00008180
IF (K.NE.KO) SUPH=SU*PHI(NKA) 00008190
RES=-B*PHI(NIB)-D*PHI(NJB)-E*PHI(N)-F*PHI(NJA)-H*PHI(NIA)-SUPH-WEL00008200
1L(N)-RHO*OLD(N)-QR-SL(L)
V(N)=(RES-BL*V(NIB)-CL*V(NJB))/DL 00008210
150 CONTINUE 00008220
J9=J6 00008230

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

152 J4=J5+1          0000824
155 CONTINUE         0000825
C
C   ---BACK SUBSTITUTE FOR VECTOR XI---
DO 165 K=1,K0        0000826
K3=K0-K+1            0000827
J4=JDIML2(I0)-JDIML1(I0)+2 0000828
KN=(K3-1)*NIJ        0000829
DO 162 I=1,I2        0000830
I3=I0-I              0000831
J5=JDIML2(I3)-JDIML1(I3)+J4 0000832
J6=NIJ-J4+1+KN      0000833
J7=J6+JDIML2(I3+1)-JDIML1(I3+1)+1 0000834
DO 160 IJ=J4,J5      0000835
N=NIJ-IJ+1+KN       0000836
IF(S(N).LT.0.) GO TO 160 0000837
IF(ITL.NE.ICHK(12)) GO TO 157 0000838
L=N                  0000839
LKB=N-NIJ           0000840
GO TO 158            0000841
157 L=1              0000842
LKB=1                0000843
158 IF(K3.EQ.K0) GO TO 151 0000844
IF(K3.EQ.1) GO TO 159 0000845
IF(T(N).EQ.0..AND.TK(N).EQ.0..AND.TK(N-NIJ).EQ.0..AND.TLK(L).EQ.
1 0..AND.TLK(LKB).EQ.0.) GO TO 160 0000846
GO TO 156            0000847
151 IF(T(N).EQ.0..AND.TK(N-NIJ).EQ.0..AND.TLK(LKB).EQ.0.) GO TO 160 0000848
GO TO 156            0000849
159 IF(T(N).EQ.0..AND.TK(N).EQ.0..AND.TLK(L).EQ.0.) GO TO 160 0000850
156 GLXI=0.D0         0000851
  NJA=N+1             0000852
  NIA=N+JDIML2(I3)-JDIML1(I3+1)+1 0000853
  IF(N.EQ.J6) NJA=1 0000854
  IF(NIA.GT.J7)NIA=1 0000855
  IF(NIA.LE.J6) NIA=1 0000856
  IF(I.EQ.1) NIA=1 0000857
  IF (K3.NE.K0) GLXI=GL(N)*V (N+NIJ) 0000858
  V (N)=V(N)-EL(N)*V (NJA)-FL(N)*V (NIA)-GLXI 0000859
C
C   ---COMPARE MAGNITUDE OF CHANGE WITH CLOSURE CRITERIA---
TCHK=ABS( V(N) )      0000860
IF (TCHK.GT.BIG) BIG=TCHK 0000861
PHI(N)=PHI(N)+ V(N) 0000862
XI(N)=XI(N)+ V(N) 0000863
160 CONTINUE           0000864
162 J4=J5+1            0000865
165 CONTINUE           0000866
  IF (BIG.GT.ERR) TEST=1. 0000867
  TEST3(IT+1)=BIG      0000868
  IF (TEST.EQ.0.) RETURN 0000869

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

GO TO 60	00008700
.....	.....
170 J8=1	00008710
DO 205 KK=1,K0	00008720
K=K0-KK+1	00008730
J4=JDIML2(I0)-JDIML1(I0)+2	00008740
KN=(K-1)*NIJ	00008750
DO 202 II=1,I2	00008760
I=I0-II	00008770
J5=JDIML2(I)-JDIML1(I)+J4	00008780
J6=NIJ-J5+1+KN	00008790
J7=NIJ-J4+1+KN	00008800
J9=J6-JDIML2(I-1)+JDIML1(I-1)-1	00008810
DO 200 IJ=J4,J5	00008820
N=NIJ-J5-J4+1+IJ+KN	00008830
---SKIP COMPUTATIONS IF NODE CONSTANT HEAD---	
IF (S(N).LT.0.) GO TO 200	00008840
NIB=N-JDIML2(I-1)+JDIML1(I)-1	00008850
NIA=N+JDIML2(I)-JDIML1(I+1)+1	00008860
NJA=N+1	00008870
NJB=N-1	00008880
IF(N.EQ.J6) NJB=1	00008890
IF(N.EQ.J7) NJA=1	00008900
IF(NIB.GE.J6) NIB=1	00008910
IF(NIB.LT.J9) NIB=1	00008920
IF(NIA.GT.J8) NIA=1	00008930
IF(NIA.LE.J7) NIA=1	00008940
IF(I.EQ.2) NIB=1	00008950
IF(I.EQ.I1) NIA=1	00008960
NKA=N+NIJ	00008970
NKB=N-NIJ	00008980
IF (ITL.NE.ICHK(12)) GO TO 171	00008990
L=N	00009000
LKB=NKB	00009010
GO TO 172	00009020
171 L=1	00009030
LKB=1	00009040
TLK(LKB)=0.	00009050
TLK(L)=0.	00009060
TL(L)=0.	00009070
SL(L)=0.	00009080
---SKIP COMPUTATIONS IF NODE OUTSIDE FLOW SYSTEM---	
172 IF(K.EQ.1) GO TO 164	00009090
IF(K.EQ.K0) GO TO 163	00009100
IF(T(N).EQ.0..AND.TK(N).EQ.0..AND.TK(NKB).EQ.0..AND.TLK(L).EQ.0.	00009110
1..AND.TLK(LKB).EQ.0..) GO TO 200	
GO TO 161	00009120
164 IF(T(N).EQ.0..AND.TK(N).EQ.0..AND.TLK(L).EQ.0..) GO TO 200	00009130
GO TO 161	00009140

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

C 163 IF(T(N).EQ.0..AND.TK(NKB).EQ.0..AND.TLK(LKB).EQ.0.) GO TO 200      0000915
C     ---COMPUTE COEFFICIENTS---
C 161 J=N-J6+JDIML1(I)                                              0000916
C     D=TR(NJB)/DELX(J)                                              0000917
C     F=TR(N)/DELX(J)                                               0000918
C     B=TC(NIB)/DELY(I)                                              0000919
C     H=TC(N)/DELY(I)                                               0000920
C     SU=0.DO                                                       0000921
C     Z=0.DO                                                       0000922
C     IF(K.EQ.1) GO TO 174                                           0000923
C     Z=TK(NKB) + TLK(LKB)                                         0000924
C     IF(IEQN.EQ.ICHK(11)) Z=Z/DELZ(K)                                0000925
174 IF(K.EQ.K0) GO TO 175                                           0000926
C     SU=TK(N) + TLK(L)                                             0000927
C     IF(IEQN.EQ.ICHK(11))SU=SU/DELZ(K)                                0000928
175 RHO=S(N)/DELT                                              0000929
C     QR=0.                                                       0000930
C     IF (K.EQ.K0.AND.IQRE.EQ.ICHK(7)) QR=QRE(N-KN)                  0000931
C
C     ---SIP REVERSE ALGORITHM---
C     ---FORWARD SUBSTITUTE, COMPUTING INTERMEDIATE VECTOR V---
C     IF (K.EQ.1) GO TO 181                                           0000932
C     IF (K.EQ.K0) GO TO 183                                           0000933
C     GO TO 180                                                       0000934
181 E=-B-D-F-H-SU-Z-RHO-TL(L)+TLK(L)                                0000935
C     GO TO 182                                                       0000936
183 E=-B-D-F-H-SU-Z-RHO+TLK(LKB)-TL(L)                                0000937
C     GO TO 182                                                       0000938
180 E=-B-D-F-H-SU-Z-RHO+TLK(LKB)+TLK(L)-TL(L)                                0000939
182 CONTINUE                                              0000940
C     BL=H/(1.+W*(EL(NIA)+GL(NIA)-))                                0000941
C     CL=D/(1.+W*(FL(NJB)+GL(NJB)))                                0000942
C     C=BL*EL(NIA)                                              0000943
C     G=CL*FL(NJB)                                              0000944
C     WU=CL*GL(NJB)                                              0000945
C     U=BL*GL(NIA)                                              0000946
C     IF (K.EQ.K0) GO TO 190                                           0000947
C     AL=SU/(1.+W*(EL(NKA)+FL(NKA)))                                0000948
C     A=AL*EL(NKA)                                              0000949
C     TU=AL*FL(NKA)                                              0000950
C     DL=E+W*(C+G+A+WU+TU+U)-AL*GL(NKA)-BL*FL(NIA)-CL*EL(NJB)    0000951
C     EL(N)=(F-W*(C+A))/DL                                         0000952
C     FL(N)=(B-W*(G+TU))/DL                                         0000953
C     GL(N)=(Z-W*(WU+U))/DL                                         0000954
C     ZPHI=0.DO                                              0000955
C     IF (K.NE.1) ZPHI=Z*PHI(NKB)                                 0000956
C     RES=-B*PHI(NIB)-D*PHI(NJB)-E*PHI(N)-F*PHI(NJA)-H*PHI(NIA)-SU*PHI(N) 0000957
C     1KA)-ZPHI-WELL(N)-RHO*OLD(N)-QR-SL(L)
C     V(N)=(RES-AL*V(NKA)-BL*V(NIA)-CL*V(NJB))/DL                0000958
C     GO TO 200                                              0000959
190 DL=E+W*(C+G+WU+U)-BL*FL(NIA)-CL*EL(NJB)                            0000960

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

EL(N)=(F-W*C)/DL          00009610
FL(N)=(B-W*G)/DL          00009620
GL(N)=(Z-W*(WU+U))/DL     00009630
ZPHI=0.DO                  00009640
IF (K.NE.1) ZPHI=Z*PHI(NKB) 00009650
RES=-B*PHI(NIB)-D*PHI(NJB)-E*PHI(N)-F*PHI(NJA)-H*PHI(NIA)-ZPHI-WEL 00009660
1L(N)-RHO*OLD(N)-QR-SL(L)
V(N)=(RES-BL*V(NIA)-CL*V(NJB))/DL          00009670
200    CONTINUE              00009680
      J8=J7                  00009690
202    J4=J5+1                00009700
205    CONTINUE              00009710

C
C ---BACK SUBSTITUTE FOR VECTOR XI---
      J8=1                  00009720
DO 215 K=1,K0              00009730
      J4=JDIML2(1)-JDIML1(1) + 2 00009740
      KN=(K-1)*NIJ             00009750
DO 212 I=2,I1              00009760
      J5=JDIML2(I)-JDIML1(I)+J4 00009770
      J6=J5+KN                00009780
      J7=J4+KN                00009790
DO 210 IJ=J4,J5              00009800
      N=J5-IJ+J4+KN           00009810
IF (S(N).LT.0.) GO TO 210   00009820
IF(ITL.NE.ICHK(12)) GO TO 204 00009830
      L=N                  00009840
      LKB=N-NIJ              00009850
      GO TO 206              00009860
204    L=1                  00009870
      LKB=1                 00009880
206    IF(K.EQ.1) GO TO 201   00009890
      IF(K.EQ.K0) GO TO 207   00009900
      IF(T(N).EQ.0..AND.TK(N).EQ.0..AND.TK(N-NIJ).EQ.0..AND.TLK(L).EQ.0.00009910
1 .AND.TLK(LKB).EQ.0.) GO TO 210          00009920
      GO TO 203              00009930
201    IF(T(N).EQ.0..AND.TK(N).EQ.0..AND.TLK(L).EQ.0.) GO TO 210          00009940
      GO TO 203              00009950
207    IF(T(N).EQ.0..AND.TK(N-NIJ).EQ.0..AND.TLK(LKB).EQ.0.) GO TO 210          00009960
203    GLXI=0.DO              00009970
      NJA=N+1                00009980
      NIB=N-JDIML2(I-1)+JDIML1(I)-1 00009990
      IF(NIB.GE.J7) NIB=1        00010000
      IF(NIB.LT.J8) NIB=1        00010010
      IF(I.EQ.2) NIB=1          00010020
      IF(N.EQ.J6) NJA=1          00010030
      IF (K.NE.1) GLXI=GL(N)*V (N-NIJ) 00010040
      V (N)=V(N)-EL(N)*V (NJA)-FL(N)*V (NIB)-GLXI 00010050

C
C ---COMPARE MAGNITUDE OF CHANGE WITH CLOSURE CRITERIA---
      TCHK=ABS( V(N))          00010050

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

IF (TCHK.GT.BIG) BIG=TCHK          00010060
PHI(N)=PHI(N)+ V(N)              00010070
XI(N)= XI(N)+ V(N)              00010080
210  CONTINUE                      00010090
    J8=J7                          00010100
212  J4=J5+1                       00010110
215  CONTINUE                      00010120
    IF (BIG.GT.ERR) TEST=1.          00010130
    TEST3(IT+1)=BIG                00010140
    IF (TEST.EQ.0.) RETURN          00010150
    GO TO 60                        00010160
C ..... .
C
C ---FORMATS---
C
C
220 FORMAT ('OEXCEEDED PERMITTED NUMBER OF ITERATIONS'/' ',39('*')) 00010170
230 FORMAT (///1H0,I5,22H ITERATION PARAMETERS:,1P6E15.7(/28X,
.1P6E15.7/))               00010180
240 FORMAT ('-',44X,'SOLUTION BY THE STRONGLY IMPLICIT PROCEDURE'/45X,00010190
143(' '))
    END                           00010200
    SUBROUTINE CHECKI(PHI,STRT,OLD,T,S,TR,TC,TK,WELL,DELX,DELY,DELZ,FA
1CT,JFLO,FLOW,QRE,XI,RATE,ZCB,SS,TL,TLK,SL,RM,JDIML1,JDIML2)
C -----
C COMPUTE A VOLUMETRIC BALANCE
C -----
C
C SPECIFICATIONS:
C
REAL *8PHI                         00010220
C
DIMENSION PHI(IOJO,K0), STRT(IOJO,K0), OLD(IOJO,K0), T(IOJO,K0) 00010230
1, S(IOJO,K0), TR(IOJO,K0), TC(IOJO,K0), TK(IKJK,K5), WELL(10
2JO,K0), DELX(JO), DELY(IO), DELZ(K0), FACT(K0,3), JFLO(NCH,3), FLO
3W(NCH), QRE(IQJQ)
4, XI(IOJO,K0), RATE(1), ZCB(1), SS(1), TL(1), TLK(1), SL(1), RM(1)
5, JDIML1(IO), JDIML2(IO)
C
COMMON /INTEGR/ IO,JO,K0,I1,J1,K1,NPER,KTH,ITMAX,LENGTH,KP,NWEL, 00010240
1NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCH,
2IDK1,IDK2,IWATER,IQRE,IPJP,IQJQ,IKJK,K5,IPU1,IPU2,ITK,IEQN,MODE
.,ITL,IOJO,K6,K7,ILJL
COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,QR,DELM1        00010250
COMMON /SARRAY/ ICHK(13)                                         00010260
COMMON /CK/ EFLXT,STORT,QRET,CHST,CHDT,FLUXT,PUMPT,CFLUXT,FLXNT 00010270
RETURN                                00010280
C *****
C ENTRY CHECK                           00010290
C *****

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

---INITIALIZE VARIABLES---

```

NIJ = IOJO          00010300
PUMP=0.            00010310
STOR=0.            00010320
FLUXS=0.0          00010330
CHD1=0.0          00010340
CHD2=0.0          00010350
QREFLX=0.          00010360
CFLUX=0.          00010370
FLUX=0.            00010380
ETFLUX=0.          00010390
FLXN=0.0          00010400
II=0               00010410
ACHD1=0.            00010420
ACHD2=0.            00010430
.....
```

---COMPUTE RATES, STORAGE AND PUMPAGE FOR THIS STEP---

```

DO 225 K=1,K0      00010440
  J7=1              00010450
  NK = IOJO*(K-1)   00010460
    J4=JDIML2(1)-JDIML1(1) + 2 00010470
  DO 222 I=2,I1     00010480
    J5=JDIML2(I)-JDIML1(I)+J4  00010490
    J6=J5+JDIML2(I+1)-JDIML1(I+1) + 1 00010500
    DO 220 IJ=J4,J5   00010510
      NI = IJ + NK    00010520
      NIJB=NI-NIJ     00010530
      IF(ITL.EQ.ICHK(12)) GO TO 4 00010540
      NI=1             00010550
      NIJB=1           00010560
4 IF(K.NE.1) GO TO 5 00010570
  IF (T(IJ,K).EQ.0..AND.TK(IJ,K).EQ.0..AND.TLK(NI).EQ.0.) GO TO 220 00010580
  GO TO 7             00010590
5 IF(T(IJ,K).EQ.0..AND.TK(IJ,K).EQ.0..AND.TK(IJ,K-1).EQ.0..AND. 00010600
  1 TLK(NI).EQ.0..AND.TLK(NIJB).EQ.0.) GO TO 220
7 J=IJ-J4+JDIML1(I) 00010610
  AREA=DELX(J)*DELY(I) 00010620
  VOLUME=AREA*DELZ(K) 00010630
  IF (S(IJ,K).GE.0.) GO TO 180 00010640
```

---COMPUTE FLOW RATES TO AND FROM CONSTANT HEAD BOUNDARIES---

```

II=II+1            00010650
FLOW(II)=0.         00010660
JFLO(II,1)=K       00010670
JFLO(II,2)=I       00010680
JFLO(II,3)=J       00010690
  IF(IJ.EQ.J4) GO TO 30 00010700
IF (S(IJ-1,K).LT.0..OR.T(IJ-1,K).EQ.0.) GO TO 30 00010710
X=(PHI(IJ,K)-PHI(IJ-1,K))*TR(IJ-1,K)*DELY(I) 00010720
IF(IEQN.EQ.ICHK(11)) X=X*DELZ(K) 00010730
```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

FLOW(II)=FLOW(II)+X          00010740
IF (X) 10,30,20             00010750
10 CHD1=CHD1+X              00010760
GO TO 30                   00010770
20 CHD2=CHD2+X              00010780
30 JF(IJ.EQ.J5) GO TO 60   00010790
  IF (S(IJ+1,K).LT.0..OR.T(IJ+1,K).EQ.0.) GO TO 60 00010800
  X=(PHI(IJ,K)-PHI(IJ+1,K))*DELY(I)*TR(IJ,K)      00010810
  IF(IEQN.EQ.ICHK(11)) X=X*DELZ(K)                  00010820
  FLOW(II)=FLOW(II)+X          00010830
  IF (X) 40,60,50             00010840
40 CHD1=CHD1+X              00010850
GO TO 60                   00010860
50 CHD2=CHD2+X              00010870
60 IF (K.EQ.1) GO TO 90    00010880
  IF (S(IJ,K-1).LT.0..OR.TK(IJ,K-1).EQ.0.) GO TO 90 00010890
  X=(PHI(IJ,K)-PHI(IJ,K-1))*TK(IJ,K-1)*AREA        00010900
  FLOW(II)=FLOW(II)+X          00010910
  IF (X) 70,90,80             00010920
70 CHD1=CHD1+X              00010930
GO TO 90                   00010940
80 CHD2=CHD2+X              00010950
90 IF (K.EQ.K0) GO TO 120   00010960
  IF (S(IJ,K+1).LT.0..OR.TK(IJ,K).EQ.0.) GO TO 120 00010970
  X=(PHI(IJ,K)-PHI(IJ,K+1))*TK(IJ,K)*AREA        00010980
  FLOW(II)=FLOW(II)+X          00010990
  IF (X) 100,120,110           0001100
100 CHD1=CHD1+X             0001101
GO TO 120                  0001102
110 CHD2=CHD2+X             0001103
120 NIB=IJ-JDIML2(I-1)+JDIML1(I)-1          0001104
  IF(NIB.GE.J4) NIB=1          00011050
  IF(NIB.LT.J7) NIB=1          00011060
  IF(I.EQ.2) NIB=1            00011070
  IF (S(NIB,K).LT.0..OR.T(NIB,K).EQ.0.) GO TO 150 00011080
  X=(PHI(IJ,K)-PHI(NIB,K))*TC(NIB,K)*DELX(J)      00011090
  IF(IEQN.EQ.ICHK(11)) X=X*DELZ(K)                  00011100
  FLOW(II)=FLOW(II)+X          00011110
  IF (X) 130,150,140           00011120
130 CHD1=CHD1+X             00011130
GO TO 150                  00011140
140 CHD2=CHD2+X             00011150
150 NIA=IJ+JDIML2(I)-JDIML1(I+1)+1          00011160
  IF(NIA.GT.J6) NIA=1          00011170
  IF(NIA.LE.J5) NIA=1          00011180
  IF(I.EQ.I1) NIA=1            00011190
  IF (S(NIA,K).LT.0..OR.T(NIA,K).EQ.0.) GO TO 171 00011200
  X=(PHI(IJ,K)-PHI(NIA,K))*TC(IJ,K)*DELX(J)      00011210
  IF(IEQN.EQ.ICHK(11)) X=X*DELZ(K)                  00011220
  FLOW(II)=FLOW(II)+X          00011230
  IF (X) 160,171,170           00011240

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

160 CHD1=CHD1+X          00011250
    GO TO 171            00011260
170 CHD2=CHD2+X          00011270
171 IF(FLOW(IJ)) 172,220,174 00011280
172 ACHD1=ACHD1 + AREA   00011290
    GO TO 220            00011300
174 ACHD2=ACHD2 + AREA   00011310
    GO TO 220            00011320

    ---CHECK FOR EQUATION BEING SOLVED---
180 IF(IEQN.EQ.ICHK(11)) GO TO 211           00011330

    ---EQUATION 4---
    ---RECHARGE AND WELLS---
IF (K.EQ.KO.AND.IQRE.EQ.ICHK(7)) QREFLX=QREFLX+QRE(IJ)*AREA 00011340
IF (WELL(IJ,K)) 190,210,200           00011350
190 PUMP=PUMP+WELL(IJ,K)*AREA        00011360
    GO TO 210            00011370
200 CFLUX=CFLUX+WELL(IJ,K)*AREA      00011380

    ---COMPUTE VOLUME FROM STORAGE---
210 STOR=STOR-S(IJ,K)*XI(IJ,K)*AREA     00011390
    GO TO 215            00011400

    ---EQUATION 3---
    ---RECHARGE AND WELLS---
211 IF (K.EQ.KO.AND.IQRE.EQ.ICHK(7)) QREFLX=QREFLX+QRE(IJ)*VOLUME 00011410
IF (WELL(IJ,K)) 212,214,213           00011420
212 PUMP=PUMP+WELL(IJ,K)*VOLUME       00011430
    GO TO 214            00011440
213 CFLUX=CFLUX+WELL(IJ,K)*VOLUME     00011450

    ---COMPUTE VOLUME FROM STORAGE---
214 STOR=STOR-S(IJ,K)*XI(IJ,K)*VOLUME 00011460
215 IF (ITL.NE.ICHK(12)) GO TO 220     00011470
    Z = 0.0              00011480
    SU = 0.0              00011490
    IF (K.NE.1) Z = TLK(NI-NIJ)*PHI(IJ,K-1) 00011500
    IF (K.NE.KO) SU = TLK(NI)*PHI(IJ,K+1) 00011510
    QLEAKN=-(TL(NI)*PHI(IJ,K)-SU-Z-SL(NI))*AREA 00011520
    FLUXS = FLUXS + QLEAKN      00011530
    IF (QLEAKN.LT.0) FLXN=FLXN-QLEAKN 00011540
220 CONTINUE                         00011550
    J7=J4              00011560
222 J4=J5+1                          00011570
225 CONTINUE                         00011580
    .....               00011590
    .....               00011600

    ---COMPUTE CUMULATIVE VOLUMES, TOTALS, AND DIFFERENCES---
FLUXT = FLUXT + FLUXS*DELT          00011590
FLXNT=FLXN*DELT+FLXNT             00011600

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

FLXPT = FLUXT + FLXNT          00011610
STORT=STORT+STOR              00011620
STOR=STOR/DELT                00011630
QRET=QRET+QREFLX*DELT         00011640
CHDT=CHDT-CHD1*DELT          00011650
CHST=CHST+CHD2*DELT          00011660
PUMPT=PUMPT-PUMP*DELT        00011670
CFLUXT=CFLUXT+CFLUX*DELT    00011680
TOTL1=STORT+QRET+CFLUXT+CHST+FLXPT 00011690
TOTL2=CHDT+PUMPT+ETFLXT+FLXNT   00011700
SUMR=QREFLX+CFLUX+CHD2+CHD1+PUMP+ETFLUX+FLUXS+STOR 00011710
DIFF=TOTL2-TOTL1              00011720
PERCNT=0.0                     00011730
IF (TOTL2.EQ.0.) GO TO 230      00011740
PERCNT=DIFF/TOTL2*100.          00011750
230 RETURN                      00011760
C ..... .
C
C ---PRINT RESULTS---
C *****
ENTRY CWRITE                   00011770
C *****
C
C      WRITE (6,260) STOR,QREFLX,STORT,CFLUX,QRET,PUMP,CFLUXT,ETFLUX,CHST
C      1,FLXPT,CHD2,TOTL1,CHD1,FLUX,FLUXS,ETFLXT,CHDT,SUMR,PUMPT,FLXNT,TOT
C      2L2,DIFF,PERCNT           00011780
C
C ---COMPUTE VERTICAL FLOW---
240 X=0.                         00011790
Y=0.                           00011800
IF (K0.EQ.1) RETURN             00011810
J4=JDIML2(1)-JDIML1(1) + 2     00011820
DO 252 I=2,I1                  00011830
J5=JDIML2(I)-JDIML1(I)+J4     00011840
DO 250 IJ=J4,J5                00011850
J=IJ-J4+JDIML1(I)              00011860
X=X+(PHI(IJ,1)-PHI(IJ,2))*TK(IJ,1)*DELX(J)*DELY(I) 00011870
Y=Y+(PHI(IJ,K1)-PHI(IJ,K0))*TK(IJ,K1)*DELX(J)*DELY(I) 00011880
250  CONTINUE                   00011890
252  J4=J5+1                   00011900
      WRITE (6,290) Y,X          00011910
      WRITE (6,300) ACHD1,ACHD2  00011920
      RETURN                      00011930
C
C ---FORMATS---
C
C -----
C
C
260 FORMAT ('0',10X,'CUMULATIVE MASS BALANCE:',16X,'L**3',23X,'RATES F00011940

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

10R THIS TIME STEP:',16X,'L**3/T'/11X,24('-'),43X,25('-')//20X,'SOU
•2RCES:',69X,'STORAGE =',F20.4/20X,8('-'),68X,'RECHARGE =',F20.4/27X
3,'STORAGE =',F20.2,35X,'CONSTANT FLUX =',F20.4/26X,'RECHARGE =',F2
40.2,41X,'PUMPING =',F20.4/21X,'CONSTANT FLUX =',F20.2,30X,'EVAPOTR
5ANSPIRATION =',F20.4/21X,'CONSTANT HEAD =',F20.2,34X,'CONSTANT HEA
6D:/27X,'LEAKAGE =',F20.2,46X,'IN =',F20.4/21X,'TOTAL SOURCES =',F
720.2,45X,'OUT =',F20.4/96X,'LEAKAGE:/20X,'DISCHARGES:',45X,'FROM
8PREVIOUS PUMPING PERIOD =',F20.4/20X,11('-'),68X,'TOTAL =',F20.4/1
96X,'EVAPOTRANSPIRATION =',F20.2/21X,'CONSTANT HEAD =',F20.2,36X,'S
nUM OF RATES =',F20.4/19X'QUANTITY PUMPED =',F20.2/27X,'LEAKAGE =',
nF20.2/19X,'TOTAL DISCHARGE =',F20.2//17X,'DISCHARGE-SOURCES =',F20
n.2/15X,'PER CENT DIFFERENCE =',F20.2//)
290 FORMAT ('OFLow TO TOP LAYER =',G15.7,' FLOW TO BOTTOM LAYER =',G00011950
115.7,' POSITIVE UPWARD')
300 FORMAT ('0',5X,'AREA OF CONSTANT HEAD DISCHARGE = ',G15.7,'L**2', 00011960
13X,'AREA OF CONSTANT HEAD SOURCE = ',G15.7,'L**2')
310 FORMAT('0',5X,'BOUNDARY FLUX IN =',G15.7,'L**3/S',3X,'BOUNDARY FLU00011970
1X OUT =',G15.7,'L**3/S'/5X,'NOTE: THESE FLUXES ARE INCLUDED IN THE
2 BUDGET AS CONSTANT FLUX AND PUMPING')
END 00011980
SUBROUTINE COEF(PHI,STRT,OLD,T,S,TR,TC,TK,WELL,DELX,DELY,DELZ,FACT00011990
1,PERM,BOTTOM,QRE,XI,RATE,ZCB,SS,TL,TLK,SL,RM,JDIML1,JDIML2)
-----
COMPUTE COEFFICIENTS
-----

SPECIFICATIONS:
REAL *8PHI 00012000

DIMENSION PHI(IOJO,K0), STRT(IOJO,K0), OLD(IOJO,K0), T(IOJO,K0 00012010
1), S(IOJO,K0), TR(IOJO,K0), TC(IOJO,K0), TK(IKJK,K5), WELL(IO
2JO,K0), DELX(JO), DELY(IO), DELZ(K0), FACT(KO,3), PERM(IPJP), BOT
3TOM(IPJP), QRE(IQJQ)
.,XI(IOJO,K0),RATE(ILJL,K7),ZCB(ILJL,K7),TL(ILJL,K6),SS(K7),
.TLK(ILJL,K6),SL(ILJL,K6),RM(2,MODE,ILJL,K7),JDIML1(IO),JDIML2(IO)
DIMENSION A(10,10),B(10),B1(10),BO(10) 00012020

COMMON /INTEGR/ IO,JO,K0,I1,J1,K1,NPER,KTH,ITMAX,LENGTH,KP,NWEL, 00012030
1NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCH,
2IDK1, IDK2, IWATER, IQRE, IPJP, IQJQ, IKJK, K5, IPU1, IPU2, ITK, IEQN, MODE
.,ITL,IOJO,K6,K7,ILJL
COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,QR,DELM1 00012040
COMMON /SARRAY/ ICHK(13) 00012050
RETURN 00012060
-----
---COMPUTE TRANSMISSIVITY FOR UPPER HYDROLOGIC UNIT WHEN
IT IS UNCONFINED---
***** ENTRY TRANS(N3) 00012070
***** J6=1 00012080

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

J4=JDIML2(1)-JDIML1(1)+2          0001209
DO 12 I=2,I1                      0001210
    J5=JDIML2(I)-JDIML1(I)+J4      0001211
DO 10 IJ=J4,J5                    0001212
IF (PERM(IJ).EQ.0.) GO TO 10      0001213
T(IJ,K0)=PERM(IJ)*(PHI(IJ,K0)-BOTTOM(IJ)) 0001214
IF (T(IJ,K0).GT.0.) GO TO 10      0001215
    J=IJ-J4+JDIML1(I)            0001216
IF (WELL(IJ,K0).LT.0.) WRITE (6,600) I,J,K0 0001217
IF (WELL(IJ,K0).GE.0.) WRITE (6,610) I,J,K0 0001218
PERM(IJ)=0.                        0001219
T(IJ,K0)=0.                        0001220
NJB=IJ-1                          0001221
    IF(IJ.EQ.J4) NJB=1            0001222
TR(NJB,K0)=0.                      0001223
TR(IJ,K0)=0.                        0001224
TC(IJ,K0)=0.                        0001225
    NIB=IJ-JDIML2(I-1)+JDIML1(I)-1 0001226
    IF(I.EQ.2) NIB=1              0001227
    IF(NIB.GE.J4) NIB=1            0001228
    IF(NIB.LT.J6) NIB=1            0001229
TC(NIB,K0)=0.                      0001230
IF (K0.NE.1) TK(IJ,K1)=0.          0001231
PHI(IJ,K0)=1.D30                  0001232
10 CONTINUE                         0001233
    J6=J4                          0001234
12   J4=J5+1                        0001235
    IF (N3.EQ.1) RETURN             0001236
    N1=K0                          0001237
    N2=K0                          0001238
    N4=K1                          0001239
    GO TO 20                      0001240
C ..... .
C ---COMPUTE T COEFFICIENTS---
C ****
C ENTRY TCOF                         0001241
C ****
    N1=1                            0001242
    N2=K0                          0001243
    N4=1                            0001244
20 DO 45 K=N1,N2                    0001245
    J4=1                            0001246
    DO 42 I=1,I1                    0001247
        J5=JDIML2(I)-JDIML1(I)+J4  0001248
        J6=J5+JDIML2(I+1)-JDIML1(I+1) + 1 0001249
    DO 39 IJ=J4,J5                  0001250
    IF (T(IJ,K).EQ.0.) GO TO 39      0001251
    NJA=IJ+1                        0001252
    IF(NJA.EQ.J5+1) NJA=1            0001253
    J=IJ-J4+JDIML1(I)                0001254

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

IF (T(NJA,K).EQ.0.) GO TO 30	00012550
TR(IJ,K)=(2.*T(NJA,K)*T(IJ,K))/(T(IJ,K)*DELX(J+1)+T(NJA,K)*	00012560
1DELX(J))*FACT(K,1)	
30 NIA=IJ+JDIML2(I)-JDIML1(I+1)+1	00012570
IF(NIA.GT.J6) NIA=1	00012580
IF(NIA.LE.J5) NIA=1	00012590
IF(I.EQ.I1) NIA=1	00012600
IF (T(NIA,K).EQ.0.) GO TO 39	00012610
TC(IJ,K)=(2.*T(NIA,K)*T(IJ,K))/(T(IJ,K)*DELY(I+1)+T(NIA,K)*	00012620
1DELY(I))*FACT(K,2)	
39 CONTINUE	00012630
42 J4=J5+1	00012640
45 CONTINUE	00012650
IF (K0.EQ.1.OR.ITK.EQ.ICHK(10).OR.N3.EQ.0) RETURN	00012660
DO 53 K=N4,K1	00012670
J4=1	00012680
DO 52 I=2,I1	00012690
J5=JDIML2(I)-JDIML1(I)+J4	00012700
DO 50 IJ=J4,J5	00012710
IF (T(IJ,K+1).EQ.0.) GO TO 50	00012720
T1=T(IJ,K)*FACT(K,3)	00012730
T2=T(IJ,K+1)*FACT(K+1,3)	00012740
TK(IJ,K)=(2.*T2*T1)/(T1*DELZ(K+1)+T2*DELZ(K))	00012750
50 CONTINUE	00012760
52 J4=J5+1	00012770
53 CONTINUE	00012780
RETURN	00012790

.....

---COMPUTE COEFFICIENTS FOR TRANSIENT PART OF LEAKAGE TERM---

\*\*\*\*\*  
ENTRY CLAY  
\*\*\*\*\*

R24 = 1.0/24.0	00012800
P1 = 3.1415927	00012810
P2 = 9.8696044	00012820
P4 = 97.409091	00012830
F0 = P2/6.0	00012840
G0 = -P2/12.0	00012850
G1 = -7.0*P4/720.0	00012860
G2 = -0.5	00012870
DO 60 K=1,K0	00012880
DO 60 IJ=1,IOJO	00012890
TL(IJ,K) = 0.0	00012900
TLK(IJ,K) = 0.0	00012910
60 SL(IJ,K) = 0.0	00012920
IF (SS(1).GT.0.) GO TO 65	00012930
DO 55 K=1,K1	00012940
KP1 = K + 1	00012950
DO 55 IJ=1,IOJO	00012960
IF (ZCB(IJ,K).LE.0.) GO TO 55	00012970
	00012980

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

AFACT=RATE(IJ,K)/ZCB(IJ,K)          00012990
TLK(IJ,K) = AFACT                  00013000
TL(IJ,KP1) = AFACT                  00013010
TL(IJ,K) = TL(IJ,K) + AFACT        00013020
55 CONTINUE                         00013030
RETURN                               00013040
65 IF(KP.NE.1.OR.KT.NE.1) GO TO 160 00013050
IF (MODE.GT.0)   GO TO 80           00013060
B(1) = G0                           00013070
B1(1) = F0                          00013080
GO TO 160                           00013090
80 IF (MODE.GT.1)   GO TO 90         00013100
B(1) = G0                           00013110
GO TO 135                           00013120
90 DO 100 J=1,MODE                 00013130
B(J) = 0.0                          00013140
RJ2 = J*j                           00013150
DO 100 I=1,MODE                   00013160
I3 = I - 3                         00013170
100 A(I,J) = RJ2**I3              00013180
B(1) = G1                           00013190
B(2) = G0                           00013200
B(3) = G2                           00013210
DO 110 I=2,MODE                   00013220
RAIJ = 1.0/A(I-1,I-1)              00013230
DO 110 K=I,MODE                   00013240
AA = A(K,I-1)*RAIJ                00013250
B(K) = B(K)-B(I-1)*AA            00013260
DO 110 J=I,MODE                   00013270
110 A(K,J) = A(K,J) - AA*A(I-1,J) 00013280
B(MODE) = B(MODE)/A(MODE,MODE)    00013290
DO 130 K=2,MODE                   00013300
J = MODE + 2 - K                 00013310
JJ = J - 1                         00013320
BJJ = B(JJ)                        00013330
DO 120 I=J,MODE                   00013340
120 BJJ = BJJ - A(JJ,I)*B(I)      00013350
130 B(JJ) = BJJ/A(JJ,JJ)          00013360
135 SUMN2 = 0.                      00013370
SUMN4 = 0.                          00013380
DO 140 K=1,MODE                   00013390
B1(K) = 1.0                         00013400
L = MODE + 1 - K                 00013410
B(L+1) = B(L)                      00013420
SUMN4 = SUMN4 + 1.0/(K*K*K*K*1.0) 00013430
140 SUMN2 = SUMN2 + 1.0/(K*K*1.0) 00013440
CN = (P4/90.0-SUMN4)*MODE**4     00013450
B1(MODE+1) = 1.0 + CN             00013460
B1(1) = F0 - SUMN2 - CN/(1.0*MODE*MODE) 00013470
B(1) = 0.0                          00013480
DO 145 K=1,MODE                   00013490

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

RK = 1.0/(1.0*K*K)          00013500
RK = RK*RK                  00013510
B1(K+1) = RK*B1(K+1)        00013520
145 B(K+1) = RK*B(K+1)      00013530
DO 150 K=1,K1               00013540
DO 150 IJ=2,IOJO            00013550
---SKIP COMPUTATIONS IF T, RATE OR M = 0, OR IF CONSTANT
HEAD BOUNDARY---
IF (RATE(IJ,K).LE.0..OR.T(IJ,K).EQ.0..OR.ZCB(IJ,K).EQ.0..OR.
T(IJ,K+1).EQ.0.) GO TO 155          00013560
RATE(IJ,K) = RATE(IJ,K)/ZCB(IJ,K)    00013570
ZCB(IJ,K) = RATE(IJ,K)*P2/(ZCB(IJ,K)*SS(K)) 00013580
IF(IDK1.EQ.ICHK(4).OR.IPU1.EQ.ICHK(8)) GO TO 150 00013590
DO 151 M=1,MODE                00013600
RM(1,M,IJ,K) = 0.0              00013610
151 RM(2,M,IJ,K) = 0.0          00013620
GO TO 150                      00013630
155 RATE(IJ,K) = - 1.0          00013640
150 CONTINUE                    00013650
160 CONTINUE                    00013660
DO 180 K=1,K1                  00013670
DO 180 IJ=2,IOJO                00013680
---SKIP COMPUTATIONS IF T, RATE OR M = 0, OR IF CONSTANT
HEAD BOUNDARY---
IF (RATE(IJ,K).LT.0.0) GO TO 180          00013690
XLM = ZCB(IJ,K)                00013700
XLMT = XLM*DELT                00013710
RXLMT = 1.0/XLMT                00013720
XLMT1 = XLM*DELM1                00013730
RXLMT1 = 1.0/XLMT1              00013740
TLN = RATE(IJ,K)                00013750
HIM1 = PHI(IJ,K)                00013760
HIM2 = PHI(IJ,K+1)              00013770
SLN = TLN*(HIM2-HIM1)            00013780
SLM = TLN*(HIM1-HIM2)            00013790
C1 = 2.*TLN*RXLMT                00013800
C2 = 2.*TLN*RXLMT1               00013810
TLM = 0.5*( TLN - C1*P2/6.0)      00013820
TLN = 0.5*( TLN + C1*P2/3.0 )    00013830
IF (MODE.LT.1) GO TO 175          00013840
C1 = C1*RXLMT                  00013850
C2 = C2*RXLMT1*DELM1             00013860
RDELT = 1.0/DELT                 00013870
OLDM1 = XI(IJ,K)                00013880
OLDM2 = XI(IJ,K+1)              00013890
DO 170 M=1,MODE                 00013900
BM = B(M+1)                     00013910
B1M = B1(M+1)                   00013920
XN = M*M                         00013930
XPP = XN*XLMT                   00013940
TEX = XPP*(24.+XPP*(-12.+XPP*(4.-XPP)))*R24 00013950

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```

IF (XPP.GT.1.OE-01)    TEX = 1.0-EXP(-XPP)          0001396
XPN = XN*XLMT          0001397
XXN = XPN*(24.+XPN*(-12.+XPN*(4.-XPN)))*R24      0001398
IF (XPN.GT.1.OE-01)    XXN = 1.0-EXP(-XPN)          0001399
DTEX = 1.0 - XXN          0001400
XXE = C1*XXN          0001401
TLN = TLN - XXE*B1M      0001402
TLM = TLM - XXE*BM      0001403
RM(1,M,IJ,K) = RM(1,M,IJ,K)-C2*(B1M*OLDM1-BM*OLDM2)*TEX 0001404
RM(2,M,IJ,K) = RM(2,M,IJ,K)-C2*(B1M*OLDM2-BM*OLDM1)*TEX 0001405
SLN = SLN + RM(1,M,IJ,K)*XXN*RDELT      0001406
SLM = SLM + RM(2,M,IJ,K)*XXN*RDELT      0001407
RM(1,M,IJ,K) = RM(1,M,IJ,K)*DTEX          0001408
170 RM(2,M,IJ,K) = RM(2,M,IJ,K)*DTEX          0001409
175 SLN = SLN + TLN*HIM1 - TLM*HIM2          0001410
      SLM = SLM + TLN*HIM2 - TLM*HIM1          0001411
      TL(IJ,K) = TL(IJ,K) + TLN          0001412
      TL(IJ,K+1) = TLN          0001413
      TLK(IJ,K) = TLM          0001414
      SL(IJ,K+1) = SLM          0001415
      SL(IJ,K) = SL(IJ,K) + SLN          0001416
180 CONTINUE          0001417
      RETURN          0001418
600 FORMAT ('-',20('*'),'WELL',2I3,' IN LAYER',I3,' GOES DRY',20('*'))0001419
610 FORMAT ('-',20('*'),'NODE',2I3,' IN LAYER',I3,' GOES DRY',20('*'))0001420
END          0001421
BLOCK DATA          0001422
-----
SPECIFICATIONS:
COMMON /SARRAY/ ICHK(13)          0001423
COMMON /PR/ DIGIT(129),VF4(12),VF5(12),          0001424
1VF6(12),VF7(12),VF8(12),VF9(12),VF10(12)
*****
DATA ICHK/'DRAW','HEAD','MASS','DK1','DK2','WATE','RECH','PUN1','P0001425
1UN2','ITKR','EQN3','ITLR','MPTY'/
DATA DIGIT/'1','2','3','4','5','6','7','8','9','10','11','12','13'0001426
1,'14','15','16','17','18','19','20','21','22','23','24','25','26',
2'27','28','29','30','31','32','33','34','35','36','37','38','39','
340','41','42','43','44','45','46','47','48','49','50','51','52','5
43','54','55','56','57','58','59','60','61','62','63','64','65','66
5','67','68','69','70','71','72','73','74','75','76','77','78','7
79
6','80','81','82','83','84','85','86','87','88','89','90','91','92
7','93','94','95','96','97','98','99','100','101','102','103','104
8','105','106','107','108','109','110','111','112','113','114','115
9','116','117','118','119','120','121','122','','','/','/','/','/','/
./',' ','X,'/
DATA VF4/(' ',' ',' ',' ',' ',' ',' ','F10. ','4/(8 ','F10. ','4))',3 0001427
1*' '

```

APPENDIX 2  
LISTING OF SOURCE CODE--Continued

```
DATA VF5/'(/I3',' ',' ',' ','X',' ',' ','F6.1','/(5X',
1',20F','6.1)',,2*,')',/          00014280
DATA VF6/'( ',' ',' ',' ',' ',' ','F4.0','/(20','F4.0','))',3 00014290
1*' ,/
DATA VF7/'(/I6',' ',' ',' ','X',' ',' ','F9.5','/(1H',' ',5X',
1',14F','9.5)',,')',,')',/          00014300
DATA VF8/'(/I6',' ',' ',' ','X,1P',' ',' ','E12.','5/(1','H ,5'
1,'X,1P','10E1','2.5)',,')',/        00014310
DATA VF9/'(/I6',' ',' ',' ','X,1P',' ',' ','E11.','3/(1','H ,5'
1,'X,1P','10E1','1.3)',,')',/        00014320
DATA VF10/'(/I5',' ',' ',' ','X',' ',' ','F7.2','/(5X','18F'
1,'7.2',')',,2*,')',/           00014330
*****END*****                         00014340
```

☆U.S. GOVERNMENT PRINTING OFFICE: 1982 605 601 60